

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ **6.050103 «Програмна інженерія»**

на тему _____ **WEB-застосування для управління коштами**

Виконав:

студент IV курсу, групи

_____ ІП-43 Каменський Євгеній Антонович
(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

_____ асистент Очеретяний О.К.
посада, науковий ступінь, вчене звання, прізвище, ініціали

_____ (підпис)

**Консультант
з графічної
документації**

_____ доц., к.т.н., Ліщук К.І.
посада, науковий ступінь, вчене звання, прізвище, ініціали

_____ (підпис)

Рецензент:

_____ доц., к.т.н., доц. кафедри ТК Ткач М.М.
посада, науковий ступінь, вчене звання, прізвище, ініціали

_____ (підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – **6.050103**
«Програмна інженерія» (Програмне забезпечення систем)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

I.П. Муха
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Каменському Євгенію Антоновичу
(прізвище, ім'я, по батькові)

1. Тема проекту «WEB-застосування для управління коштами»

керівник проекту Очеретяний Олександр Костянтинович, асистент.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

2. Термін подання студентом проекту «3» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: загальні положення,

Змістовний опис і аналіз предметної області, аналіз успішних ІТ-проектів,

Аналіз вимог до програмного забезпечення

2) Моделювання та конструювання програмного забезпечення: моделювання та

аналіз програмного забезпечення, архітектура програмного забезпечення,

Конструювання програмного забезпечення

3) Аналіз якості та тестування програмного забезпечення, аналіз якості ПЗ, опис

контрольного прикладу тестування продукту

4) Впровадження та супровід програмного забезпечення, розгортання, робота ПЗ

5. Перелік графічного матеріалу

1) Схема структурна класів програмного забезпечення

2) Схема структурна варіантів використання

3) Схема бази даних

4) Схема структурна станів системи

5) Креслення вигляду екранних форм

6. Консультанти розділів проекту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |

7. Дата видачі завдання «12» березня 2019 року

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломного проекту | Термін виконання етапів проекту | Примітка |
|-------|--|---------------------------------|----------|
| 1. | Вивчення рекомендованої літератури | 06.02.2019 | |
| 2. | Аналіз існуючих методів розв'язання задачі | 15.02.2019 | |
| 3. | Постановка та формалізація задачі | 21.02.2019 | |
| 4. | Аналіз вимог до програмного забезпечення | 07.03.2019 | |
| 5. | Алгоритмізація задачі | 15.03.2019 | |
| 6. | Моделювання програмного забезпечення | 27.03.2019 | |
| 7. | Обґрунтування використовуваних технічних засобів | 05.04.2019 | |
| 8. | Розробка архітектури програмного забезпечення | 10.04.2019 | |
| 9. | Розробка програмного забезпечення | 25.04.2019 | |
| 10. | Налагодження програми | 10.05.2019 | |
| 11. | Виконання графічних документів | 14.05.2019 | |
| 12. | Оформлення пояснювальної записки | 19.05.2019 | |
| 13. | Подання ДП на попередній захист | 21.05.2019 | |
| 14. | Подання ДП рецензенту | 21.05.2019 | |
| 15. | Подання ДП на основний захист | 08.06.2019 | |

Студент _____ Каменський Є.А.
(підпис)

Керівник проекту _____ Очеретяний О.К.
(підпис)

[illegible]

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Технічне завдання

КПІ.ІП-4309.045440-03-91

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 року

ЗМІСТ

| | | |
|----------|--|-----------|
| 1 | НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ..... | 3 |
| 2 | ПІДСТАВА ДЛЯ РОЗРОБКИ | 4 |
| 3 | ПРИЗНАЧЕННЯ РОЗРОБКИ..... | 5 |
| 4 | ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 6 |
| 4.1 | ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК | 6 |
| 4.2 | ВИМОГИ ДО НАДІЙНОСТІ | 7 |
| 4.3 | УМОВИ ЕКСПЛУАТАЦІЇ | 8 |
| 4.4 | ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ..... | 8 |
| 4.5 | ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ | 9 |
| 4.6 | ВИМОГИ ДО МАРКУВАННЯ ТА ПАКУВАННЯ..... | 9 |
| 4.7 | ВИМОГИ ДО ТРАНСПОРТУВАННЯ ТА ЗБЕРІГАННЯ | 9 |
| 4.8 | СПЕЦІАЛЬНІ ВИМОГИ..... | 9 |
| 5 | ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ..... | 10 |
| 6 | СТАДІЇ І ЕТАПИ РОЗРОБКИ..... | 11 |
| 7 | ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ..... | 12 |

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: WEB-застосування для управління коштами.

Галузь застосування:

Наведене технічне завдання поширюється на розробку веб-додатку для управління коштами [045440], котра використовується для управління користувачькими грошима, їх збереження на рахунках, конвертування в різні валюти, призначена для людей які зацікавленні в збереженні та конвертації грошей з національної валюти в валюти інших країн та криптовалюти і навпаки.

Веб-застосування має вікові обмеження згідно з чинним законодавством країни, о праві відкриття та використання банківських рахунків.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.03.91 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки WEB-застосування для управління коштами є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.03.91 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для управління коштами користувачів, надання функціоналу для переведення коштів із різних рахунків, та інформування користувача про його поточний баланс та історії надходжень та витрат.

Метою розробки є вдосконалення методів керування коштами для приватної особи.

Додаток буде представляти собою продуманий та інтуїтивно зрозумілий сучасний інтерфейс із використанням найпопулярніших рішень.

Цілі додатку:

- надання зручного та безпечного сервісу;
- надання можливості виконувати перекази між різними рахунками;
- надання шаблонів переказів;
- надання можливості виконувати продаж та купівлю валюти;
- надання інформації про ціну на різні валюти;
- надання користувачу загальну інформацію про його баланси.

Цільовим продуктом даної розробки, буде зручний застосунок для усіх вікових категорій, для його використання достатньо мати навички володіння комп'ютером та браузером, це рішення не пострибуватиме вузько спеціалізованих знань.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для користувача:

- вхід в додаток;
- вхід через імейл та пароль;
- вхід через ОТР;
- вихід із додатку;
- реєстрація в додатку;
- редагування користувацької інформації;
- зміна імені;
- зміна місця проживання;
- зміна країни;
- зміна паролю;
- зміна імейлу;
- зміна номеру телефона;
- зміна аватару;
- управління рахунками;
- додавання банківських карток;
- створення рахунку;
- видалення банківських карток;
- видалення рахунку;
- перегляд балансу;
- перегляд історію транзакцій;
- додавання крипто-рахунку;
- створення крипто-рахунку;

- видалення крипто-рахунку;
- переведення коштів;
- переведення між рахунками;
- переведення між картками;
- переведення між крипто-рахунку та фіатними рахунком;
- переведення між кріпто-рахунками;
- використання шаблонів платежів.

4.1.2 Розробку виконати на WEB платформі.

4.1.3 Додаткові вимоги:

- підтримка останніх версій браузерів;
- розробка інтерфейсу згідно поставлених вимог;
- додаток повинен бути адаптивним в використанні при різних розширеннях браузера.

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3. Передбачити можливість повідомляти користувача о наслідках його дій.

4.2.4. Передбачити механізм розширення системи.

4.2.5. Передбачити збій системи та механізм її оновлення.

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.2 Обслуговування

4.3.3 Обслуговуючий персонал

Не вимагається.

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

4.4.2 Мінімальна конфігурація технічних засобів:

- тип процесору - Intel core v5;
- об'єм ОЗП - 1 Гб;
- частота процесора - 3.3ГГц;
- об'єм жорсткого диску - 1Тб.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства Linux.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:

- для фото - .png, .jpg.

4.5.3 Результати повинні бути представлені в наступному форматі:

- для фото - .png, .jpg.

4.5.4 Програмне забезпечення повинно взаємодіяти із XMLHttpRequest та HTTP, HTTPS протоколами.

Середовище розробки - WEB storm.

Мова програмування - Typescript, Javascript, Nodejs.

Мова розміщення - HTML, SASS.

Технологія розробки - Angular 7.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 100 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Керівництво системного програміста

5.3.5 Керівництво адміністратора

5.3.6 Програма та методика тестування

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема структурна компонентів структур даних

5.4.2 Схема структурна варіантів використання

5.4.3 Схема структурна класів програмного забезпечення

5.4.4 Креслення вигляду екранних форм.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Таблиця 6.1 - Планування етапів розробки проекту

| № | Назва етапу | Строк | Звітність |
|----|---|---------------|--|
| 1. | Вивчення літератури за тематикою проекту | 5.02 - 9.02 | Список вивченої літератури |
| 2. | Розробка технічного завдання | 12.02 - 23.02 | Технічне завдання |
| 3. | Аналіз вимог та уточнення специфікацій | 26.02 - 2.03 | Специфікації програмного забезпечення |
| 4. | Проектування структури програмного забезпечення, проектування компонентів | 5.03 - 16.03 | Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму ...) |
| 5. | Програмна реалізація програмного забезпечення | 19.03 - 4.04 | Тексти програмного забезпечення |
| 6. | Тестування програмного забезпечення | 7.04 - 8.05 | Тести, результати тестування |
| 7. | Розробка матеріалів текстової частини проекту | 9.04 - 14.04 | Пояснювальна записка. |
| 8. | Розробка матеріалів графічної частини проекту | 16.05 - 17.05 | Графічний матеріал проекту |
| 9. | Оформлення технічної документації проекту | 19.05 - 20.05 | Технічна документація |

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.03.91 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 12 |

Пояснювальна записка до дипломного проекту

на тему: **WEB-застосування для управління коштами**

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка до дипломного проекту містить 65 сторінки, 3 графічних матеріалів, 13 таблиці, 15 рисунків.

Об'єкт розробки: WEB-застосування для управління коштами.

Мета дипломного проекту: вдосконалення методів керування коштами для приватної особи.

В розділі «Аналіз вимог до програмного забезпечення» було проведено аналіз популярних рішень, проведено аналіз архітектурних рішень та основних модулів для розробки (pop-up, mock-api, table), проаналізовано та виділено основні функції та користувацькі вимоги. Після проведеного аналізу було поставно комплекс завдань до застосунку інтерфейсу та архітектури.

В розділі «Моделювання та конструювання програмного забезпечення» були описані всі технічні і програмні засоби, описані технології, діаграми станів, було описана архітектура, документація, структуру бази даних.

В розділі «Аналіз якості та тестування програмного забезпечення» було описано основні види та механізми тестування. Описано процедуру проведення тестів. Було протестована коректність роботи та поведінку компонентів при різних факторів впливу.

В розділі «Впровадження та супровід програмного забезпечення» було описано процедуру розгортання та описано конфігурації для налаштування. Було описано Керівництво програміста в якому детально описано процедури інсталяції та Керівництво користувача в якому було зображено усі сторінок додатку з описами процедур використання.

Пояснювальна записка складається зі вступу, 4-х розділів, висновків, переліку посилань та додатків.

Ключові слова: ДОДАТОК, СИСТЕМА, ФРЕЙМВОРК, WEB, АРХІТЕКТУРА, КРОС-БРАУЗНІСТЬ.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 5 |

ABSTRACT

Structure and scope of work. The explanatory note to the diploma project contains 65 pages, 3 graphic materials, 13 tables, 15 drawings.

Object of development: WEB-application for funds management.

The purpose of the development: improving the methods of managing the funds for an individual person.

In the section "Analysis of Software Requirements", an analysis of popular solutions was conducted, an analysis of architectural solutions and major development modules (pop-up, mock-api, table) was analyzed, key features and user requirements were analyzed and highlighted. After the analysis, we put a set of tasks for the application of the interface and architecture.

In the "Modeling and Designing Software" section, all the technical and software tools, technology described, status charts, architecture, documentation, and database structure were described.

In the section "Analysis of the quality and testing of software" described the main types and mechanisms of testing. The procedure for conducting tests is described. The correctness of the work and the behavior of the components with different factors of influence was tested.

In the "Software Implementation and Support" section, a deployment procedure has been described and configurations are described for configuring. A Programmer's Guide was described, which describes in detail the installation procedures and the User's Guide, which depicts all pages of the appendix describing the procedures for use.

The explanatory note consists of an introduction, 4 sections, conclusions, a list of references and annexes.

Keywords: APPLICATION, SYSTEM, FRAMEWORK, WEB, ARCHITECTURE, CROSS-SHOULDER.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

КПІ.ІП-4309.045440.02.81

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| | | | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗМІСТ

| | |
|--|-----------|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ..... | 9 |
| ВСТУП..... | 11 |
| 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 12 |
| 1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ | 12 |
| 1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ | 13 |
| 1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ..... | 14 |
| 1.3.1 Аналіз відомих технічних рішень | 17 |
| 1.3.2 Аналіз відомих програмних продуктів..... | 19 |
| 1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 19 |
| 1.4.1 Розроблення функціональних вимог | 26 |
| 1.4.2 Розроблення нефункціональних вимог | 33 |
| 1.4.3 Постановка комплексу завдань модулю..... | 35 |
| 1.5 ВИСНОВКИ ПО РОЗДІЛУ | 36 |
| 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 37 |
| 2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 37 |
| 2.1 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 41 |
| 2.2 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 49 |
| 2.3 АНАЛІЗ БЕЗПЕКИ ДАНИХ | 51 |
| 2.4 ВИСНОВКИ ПО РОЗДІЛУ | 52 |
| 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 53 |
| 3.1 АНАЛІЗ ЯКОСТІ ПЗ..... | 53 |
| 3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ..... | 53 |
| 3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ | 55 |
| 3.4 ВИСНОВОК ДО РОЗДІЛУ..... | 60 |
| 4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 61 |
| 4.1 РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 61 |
| 4.2 РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ | 61 |
| 4.3 ВИСНОВОК ДО РОЗДІЛУ | 62 |
| ВИСНОВКИ | 63 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 65 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

РУТ - головна сторінка сайту.

СІ - Безперервна інтеграція (СІ) - це практика розробки, яка вимагає від розробників інтегрувати код у спільне сховище кілька разів на день. Кожна реєстрація перевіряється за допомогою автоматизованої збірки, що дозволяє командам своєчасно виявляти проблеми.

Angular - основа веб-додатків з відкритим вихідним кодом на основі TypeScript, очолювана Angular Team.

UX - Дизайн для користувача - це процес створення продуктів, які надають користувачам осмислений та відповідний досвід. Це передбачає розробку всього процесу придбання та інтеграції продукту, включаючи аспекти брендингу, дизайну, зручності використання та функціонування.

UI - В інформаційних технологіях користувацький інтерфейс - це все, що розроблено в інформаційному пристрої, з яким людина може взаємодіяти. Це може включати екрани дисплеїв, клавіатури, мишу та зовнішній вигляд робочого столу. Це також спосіб, через який користувач взаємодіє з додатком або веб-сайтом.

QR-код - (скорочено з коду швидкого реагування) є торговельною маркою для матричного штрих-коду (або двовірного штрих-коду), розробленого в 1994 році для автомобільної промисловості Японії. Штрих-код - це машиночитана оптична мітка, яка містить інформацію про предмет, до якого вона прикріплена.

ОТР - Одноразовий пароль (ОТР), також відомий як одноразовий PIN-код, є паролем, який дійсний лише для одного сеансу входу або транзакції на комп'ютерній системі або іншому цифровому пристрої.

Криптовалюта (або криптова валюта) - це цифровий актив, призначений для роботи в якості засобу обміну, який використовує сильну криптографію для забезпечення фінансових операцій, контролю за створенням додаткових одиниць і перевірки передачі активів.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| | | | | | | 9 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

CVV, CSC - Код безпеки картки, це функція безпеки для операцій з платіжними картками, які не мають картки, створених для зменшення випадків шахрайства з кредитними картами. CSC є доповненням до номера банківської картки, який вибитий або надрукований на картці. CSC використовується як функція захисту в ситуаціях, коли PIN-код не можна використовувати.

Фіатні гроші - (англ. fiat currency), також фідуціарні гроші - є законним платіжним засобом, вартість якого підтримується урядом, який його видав. Долар США - це грошові кошти, як і євро та багато інших великих світових валют. Цей підхід відрізняється від грошей, вартість яких підкріплюється деякими фізичними благами, такими як золото або срібло, які називаються товарними грошима.

Full HD 1080p - це набір відео режимів HDTV високої чіткості, які характеризуються 1920 пікселями, які відображаються по екрану горизонтально, а 1080 пікселів - по вертикалі; p означає прогресивну розгортку, тобто не переплетену.

Репрезентативна передача стану (REST) - це архітектурний стиль програмного забезпечення, який визначає набір обмежень, які використовуються для створення веб-служб. Веб-сервіси, які відповідають архітектурному стилю REST, називаються RESTful Web services (RWS), забезпечують взаємодію між комп'ютерними системами в Інтернеті. Веб-сервіси RESTful дозволяють системам, що запитують, отримувати доступ і керувати текстовими поданнями веб-ресурсів, використовуючи єдиний і заздалегідь визначений набір операцій без статусу.

Mock API - фейкове API, це інструмент завдяки якому ви можете негайно почати кодування і перейти на реальні веб-служби, коли вони будуть готові. Це робота в автономному режимі, без необхідності підключення до серверів та ручне керування даними.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 10 |

ВСТУП

Ми з вами живимо у світі де понад 200 країн в яких живе більше 7 мільярдів людей. Наш світ дуже великий та різний, але є річ з якою зустрічався кожен із нас — це гроші. На сьогоднішній день вплив грошей має найвищу ланку, гроші стали символом багатства та влади, гроші — це рушій який в лічені хвилини може змінити вашу думку. В наш час ми маємо безліч факторів пов'язаних із неправомірним середовищем в якому ми маємо ризик втратити власні гроші, через банкрутство банків або девальвації валюти.

Темою даної роботи є WEB-застосунок для управління користувацькими грошима для різних валют з можливістю обробки платежів.

Метою розробки є вдосконалення методів керування коштами для приватної особи, за допомогою WEB-застосунку, створення єдиного стабільного банку який може забезпечити безперервну роботу та довіру, як до національної валюти, так і до віртуальної.

Аналіз існуючих аналогів виявив, що ринок наповнений великою кількістю неякісних веб-застосунків. Вони є недосконалими, обмежені в масштабуванні, переповнені недоцільним контентом, більш того в них відсутня підтримка мобільних приладів.

Актуальність даної роботи полягає в створенні нового сервісу, для різних валют, із доступним зручним інтерфейсом та гнучким механізмом масштабування та інтегрування.

Завданням до даної роботи є розробка WEB-застосунку для фізичних осіб за для управління власними коштами в крипто та фіатному форматі. Компонентами розроблюваного додатку є WEB-застосунку та серверне застосування.

Результатом роботи є додаток який можна інтегрувати із різними фінансовими системами.

Веб-застосування має вікові обмеження згідно з чинним законодавством країни, о праві відкриття та використання банківських рахунків.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 11 |

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Для створення додатку було обрано концепцію - веб додаток, а це клієнт-сервер архітектура доступна в загальному майже для усіх пристроїв та операційних систем, комп'ютерів та мобільних телефонів, через браузер. Подібна характеристика має назву крос-браузерність та платформи незалежності.

Подібні рішення надають швидке редагування - комунікацію між клієнтом та банком, та простоту і зручність в оновленні, представленні нового функціоналу та видаленні застарілих даних.

Для роботи з клієнтами по управлінню грошима існують банки, цілі яких надати необхідні послуги, тому пропонується веб додаток. Це простий і зручний сервіс, через який можна виконувати всі комунікації з банком. Онлайн банк - це банк, що надає послуги, такі як, зберігання та перерахування грошових одиниць та додаткові маніпуляції, зручні в опрацюванні.

Подібний ресурс складається з серверного додатку, добре захищеного, який забезпечує роботу з фінансовими сервісами такими як Mastercard або Visa, та для роботи з крипто валютою. Можливо як безпосередньо працювати з кожною крипто валютою, так і використовувати API популярних кріптовіс. Другу частину цього ресурсу складає клієнт - інтерфейс додатку через який буде опрацьовуватися робота із клієнтом банку - що і є основною задачею цього рішення.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 12 |

1.2 Змістовний опис і аналіз предметної області

В світі банки надають безліч послуг, тому було проаналізовано, та обрано найважливіші програми. А саме основний інтерес користувачів в онлайн банках це:

- перегляд залишків на рахунках;
- оплата рахунків;
- переказ грошей;
- придбання або продажу іноземних валют;
- кредитні та депозитні програми.

Перегляд залишків на рахунках - це надання інформації про поточний стан залишків по кожному рахунку, перегляд історії по рахунках, транзакції та підрахування балансів по типам валют та середнього показника по основній валюті користувача.

Оплата рахунків - переведення коштів на різні державні чи приватні рахунки, з цілю оплати певних послуг, наприклад, поповнення мобільного телефону. Також, отримання додаткової інформації такі як - сума необхідна для поповнення телефону.

Переказ грошей - функціонал по переведенню на рахунки фізичних осіб, наприклад - друзів, підприємців чи оплати рахунків.

Придбання та продаж іноземних валют - це надання інформації про поточний курс по купівлі та продажу цікавих клієнту валют та механізм їх обміну. Не виключенням є інтерес до історії вальвації грошової одиниці в певний період.

Кредитні програми - надання фінансів користувачу на певних термін із поверненням під нульовий чи певних відсоток.

Депозитні програми - надання фінансів банку користувачем за для отримання певного додатку від відсотку.

1.3 Аналіз успішних ІТ-проектів

Після детального вивчення, в світі існує декілька зручних сервісів із подібним реалізованим наповненням. Найкращим із них на території України вважається Приватбанк[6].

1 Приватбанк[6] - найпопулярніший український банк.

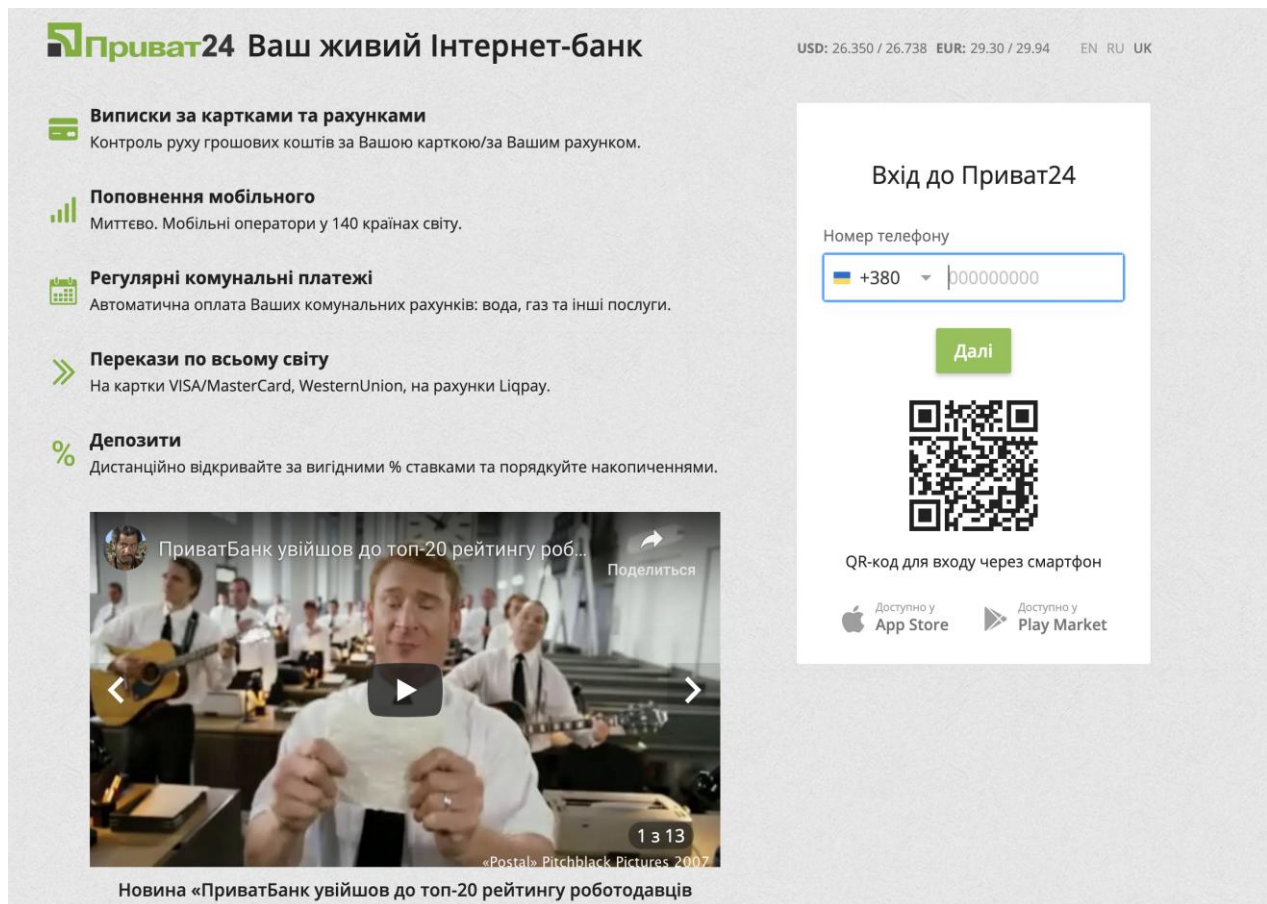


Рисунок 1.1 - Головна сторінка «Приват24»

Першим фаворитом слугує український банк Приватбанк, а саме його проект «Приват24» (рисунок 1.1). Даний сайт переповнений великим вибором усіх можливих функціонувань для банку.

Такі як:

- виписки за картками та рахунками;
- поповнення мобільного;
- регулярні комунальні платежі;
- перекази по всьому світу;

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 14 |

- депозити;
- кредити;
- магазин та багато іншого.

Недоліком даного ресурсу, я вважаю, надмірна кількість функціоналу, та перенасиченість сторінок. Великою проблемою є інтуїтивно не зрозуміле знаходження необрізного інструменту. Та місцями невдале розташування компонентів на певних сторінках, подібний функціонал не є кооперований, та такий елемент, як корзина не має необхідності в існуванні, недолік в збереженні шаблонів та їх опрацюванні, достатній дизайн, але розмитий фокус при використанні.

Друге рішення, це Сави[7] - іноземний ресурс для управління крипто фінансами (рисунок 1.2).

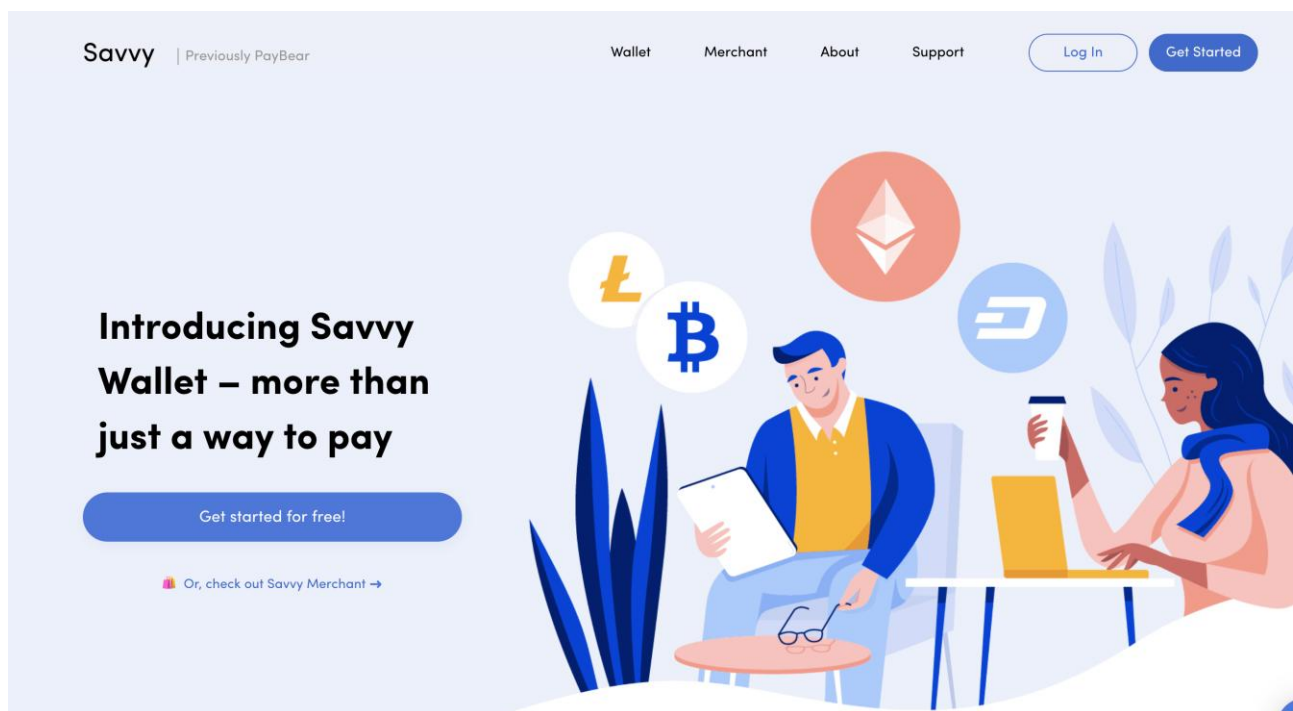


Рисунок 1.2 - Онлайн гаманець «Savvy»

Це рішення перше та найкраще в своїй сфері, це онлайн гаманець, легкий та зручний сервіс, але наміру залежить від безпеки, має підтримку декількох криптовалют, але відсутня робота із фіатними грошима, відсутній інструмент

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 15 |

оплати через шаблони, та відсутня інформація про стан валют, не є безкоштовним.

Також приставлена таблиця майже із 20 різних рішень де частково були впроваджені подібні вимоги.

Таблиця 1.1 - Відомі рішення

| Назва | Тип | Інтернет адреса |
|-----------------------------|-------------------|---|
| universalbank | Банк | https://www.universalbank.com.ua |
| monobank | Банк | https://www.monobank.com.ua |
| pumb | Банк | https://online.pumb.ua |
| digital pumb | Банк | https://www.digital.pumb.ua |
| privat client | Банк | https://www.privat24.ua |
| privat | Банк | https://privatbank.ua |
| privat bussiness user guide | Банк | https://privatbank.ua/business/internet-bank-privat24-dlya-juridicheskikh-lic |
| pumb | Банк | https://online.pumb.ua/demo-corp |
| changelly | Обмін криптовалют | https://changelly.com |
| shapeshift | Обмін криптовалют | https://shapeshift.io/#/coins |
| paritetbank | Банк | http://www.paritetbank.by/services/ |
| tas24 | Банк | https://tas24b.ua/credit/ |
| ledger | Обмін криптовалют | https://www.ledger.com |
| europeancryptobank | Обмін криптовалют | https://www.europeancryptobank.io/# |

Продовження таблиці 1.1

| | | |
|--------------|--------------------|---|
| Mycryptobank | Обмін криптовалюти | https://mycryptobank.io |
| coinpayments | Обмін криптовалюти | https://www.coinpayments.net/merchant-tools |
| cryptobank | Обмін криптовалюти | https://cryptobank.co/ |
| crypterium | Обмін криптовалюти | https://crypterium.com/ |

1.3.1 Аналіз відомих технічних рішень

Для реалізації рішення було обрано фреймворк Angular. Його перевага - це встановлена структура та простота в управлінні. Чітка опис та використання типізації Typescript робить його надійним інструментом. Це допомагає в розробці великих і складних проектів та зменшує ризики непередбачених помилок.

Angular з під коробки надає механізми написання тестів як і unit-tests, так і інтеграційних(e2e). Ще однією перевагою є нововведення - це створення бібліотек та проектів всередині одного простору. Даний підхід допомагає різним командам працювати над різними проектами одного продукту але дотримуються єдиного підходу і стилів. При порівнянні з ринком веб інструментів, на жаль, немає подібних комплексних рішень націлених на великі проекти, основним конкурентом є бібліотека React та Vue, що також дає інструмент для легкої і швидкої розробки, вони зручніше і швидше для реалізації малих і середніх веб додатків.

Для покращення роботи та якості розробки необхідно працювати із перенумерованими даними, наприклад, ресурс[3] Це гарний інструмент для розробки, має вбудований шаблон моделювання та функціонал для конфігурування. Але в ньому відсутній функціонал для debug та при наявності змін необхідно проводити перезапуск - а це є витрата по часу, тому до ряду завдань потрібно віднести розробку власного інструменту.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 17 |

Відображення повідомлень (Pop-up) - це модуль, інструмент для зображення спливаючих вікон. Він складається з 2-х частин це Alert і pop-up, Alert це встановлений модальні вікна із заздалегідь підготовленими переліком іконок і кнопок подібно ресурсу[4]. Але недостатня наявність вибору шаблонів. Ще одним недоліком - відсутність вже вбудований механізм у передачі даних в цей компонент, окрім текстових як реалізована в ресурсі [4], формату html шаблонів, що дає динамічні, багато типові модифікації і пере використання цього модуля. Також відсутній простий механізм модифікацій стилів заснований на підході іменувань bootstrap.

Друга частина даного такого модуля складається з таких компонентів які надають оболонку спрямовану на відображення динамічних даних. Крім можливості передачі html шаблонів повинно бути присутня генерація компонентів всередині оболонки pop-up, також такий компонент повинен не обмежуватися в кількості згенерованих модальних вікон. Щоб робило його гнучким у використанні для декількох спливаючи вікон. Для простоти розробки в такому модулі повинні бути присутній інтерфейс для установки конфігурації, які в подальшому будуть використовуватися як початкові значення.

Angular table - даний модуль є реалізацією сукупності багатокomпонентної архітектури, основою якої - це організація роботи з таблицями. Переважати повинно не компонент блек-бокс, а навпаки тільки атрибутивне розширення встановлених шаблонів опису таблиці в html, дана таблиця повинна не мати жорсткої ієрархічної структури, і всі взаємозв'язки повинні відбуватися через централізований сервіс або вже через описаний html. В перелік цих елементів бажано входили модифікатори на всіх рівнях, це гнучкі фільтри по 4х основних типах STRING, ENUM, DATE, NUMBER, і відображення для цих типів в комірках. Додатковим функціоналом повинно бути розширення рядків таблиць які спрямовані в 3 категорії це звичайні рядки, мобільний вид і групування даних за вказаним полем. В таблиці необхідна пагінація подібно ресурсу [5], але на відміну від цієї реалізації пагінація

повинна динамічно не перераховуватись в залежності від обраних користувачем фільтрів, Також в таблицю вельми просто розширити такими полями, як логічного формату або кастових призначених для користувача, але ресурс [5] немає такого функціоналу.

1.3.2 Аналіз відомих програмних продуктів

До прикладу UXDA[8] (рисунок 1.3), одна з провідних компаній по розробці UI та UX має великий вибір та базу розробок дизайнів для втілення в індустрії банків.

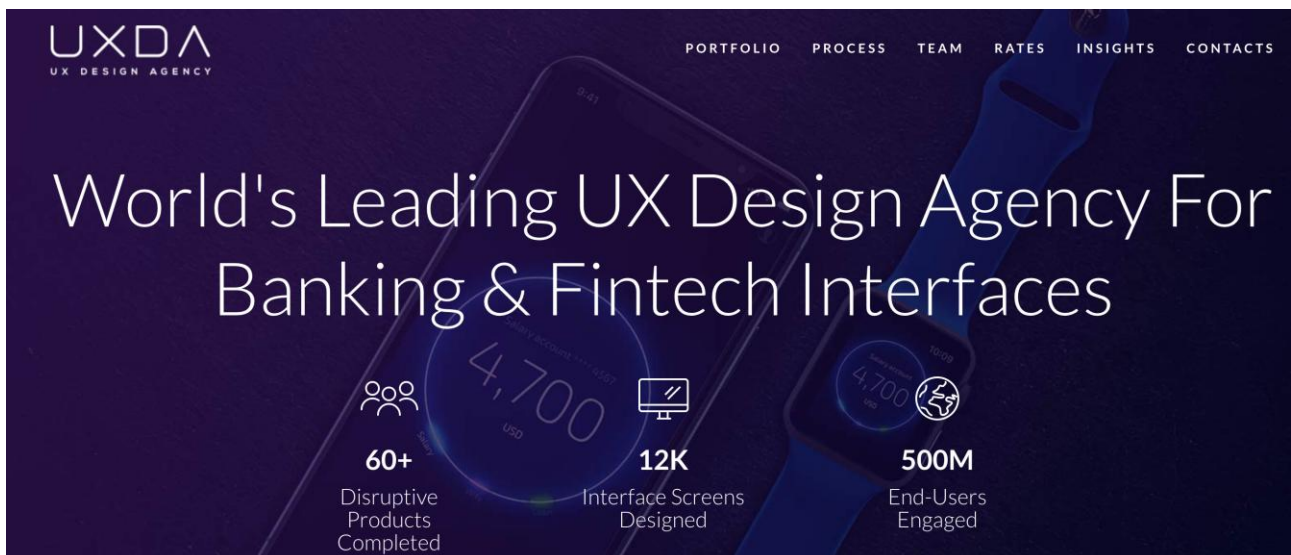


Рисунок 1.3 - Сайт компанії

Для втілення даного застосування було проведено аналіз та виявлено гнучку реалізацію сторінок платежів, вдало виконано сторінки dashboard та account. Дана реалізація розміщення компонентів інтерфейсу, приватизація цих елементів, функціонер, виконана на професійному рівні тощо.

1.4 Аналіз вимог до програмного забезпечення

Під час аналізу вимог до програмного забезпечення та вивчення ринку подібних продукція було визначено основні функції необхідні для впровадження даного додатку.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 19 |



Оскільки програма в найкращому випадку представляє автономну роботу системи без впливу на неї людського фактору, тому в системі представлення лише один актор — користувач.

Основні бізнес функції до даного продукту:

- вхід в додаток;
- вхід через імейл та пароль;
- вхід через ОТР;
- реєстрація в додатку;
- редагування користувацької інформації;
- управління рахунками;
- додавання банківських карток;
- створення рахунку;
- видалення банківських карток;
- видалення рахунку;
- перегляд балансу;
- перегляд історію транзакцій;
- додавання крипто-рахунку;
- створення крипто-рахунку;
- видалення крипто-рахунку;
- переведення коштів;
- переведення між рахунками;
- переведення між картками;
- переведення між крипто-рахунку та фіатними рахуном;
- переведення між кріпто-рахунками;
- використання шаблонів платежів.

Таблиця 1.2 – Основні варіанти використання

| ID | Актор | Назва | Передумова | Опис |
|-------|------------|---------------------------------------|------------------------------------|---|
| UC1.0 | Користувач | Вхід в додаток | Користувач має аккаунт | Користувач має можливість авторизуватися в додатку |
| UC1.1 | Користувач | Вхід через імейл та пароль | Користувач має аккаунт | Користувач використовує імейл та пароль для входу в додаток |
| UC1.2 | Користувач | Вхід через OTP | Користувач має аккаунт | Користувач використовує номер телефону та OTP для входу в додаток |
| UC2.0 | Користувач | Реєстрація в додатку | | Користувач авторизується в додатку |
| UC3.0 | Користувач | Редагування користувацької інформації | Користувач має аккаунт | Користувач може змінити власні данні |
| UC4.0 | Користувач | Управління рахунками | Користувач має аккаунт | Користувач може управляти власними рахунками |
| UC4.1 | Користувач | Додавання банківських карток | Користувач має картку іншого банку | Користувач може додати картку іншого банку |

Продовження таблиці 1.2

| | | | | |
|-------|------------|------------------------------|--|--|
| UC4.2 | Користувач | Створення карток | Користувач має аккаунт | Користувач може відкрити новий рахунок |
| UC4.3 | Користувач | Видалення банківських карток | Користувач добавив в систему банківську картку | Користувач може видалити рахунок, картку іншого банку |
| UC4.4 | Користувач | Видалення рахунку | Користувач створив рахунок | Користувач може видалили рахунок |
| UC4.5 | Користувач | Перегляд балансу | Користувач має рахунки | Користувач може переглянути поточний баланс по кожному рахунку |
| UC4.6 | Користувач | Перегляд історію транзакцій | Користувач перевів гроші | Користувач може переглянути історію транзакцій по рахункам |
| UC4.7 | Користувач | Додавання крипто-рахунку | Користувач має створений крипторахунок | Користувач може добавлять власні крипторахунки |
| UC4.8 | Користувач | Створення крипто-рахунку | Користувач має аккаунт | Користквач може замовити новій крипторахунок |

Продовження таблиці 1.2

| | | | | |
|-------|------------|---|---|---|
| UC4.9 | Користувач | Видалення крипто-рахунку | Користувач повинен мати крипторахунок | Користувач може видалити власний крипторхізм із системи |
| UC5.0 | Користувач | Переведення коштів | Користувач повинен мати відкриті ранунки | Користувач може переводити гроші між різними рахунками |
| UC5.1 | Користувач | Переведення між картками | Користувач повинен мати фіатний рахунок із коштами прив'язаний до певної картки | Користувач може перевести гроші з картки на картку за допомогою CVV |
| UC5.2 | Користувач | Переведення між рахунками | Користувач повинен мати фіатний рахунок | Користувач може перевести гроші з власного ранку на рахунок іншого банку |
| UC5.3 | Користувач | Переведення між крипто-рахунку та фіатними рахуном | Користувач повинен мати фіатний та крипто рахунок | Користувач може вивести гроші із крипт-рахунку на власний рахунок |

Продовження таблиці 1.2

| | | | | |
|-------|------------|----------------------------------|--|---|
| UC5.4 | Користувач | Переведення між крипто-рахунками | Користувач повинен мати крипто рахунок | Користувач може перевести гроші з одного крипто-рахунку на інший крипто-рахунок |
| UC6.0 | Користувач | Використання шаблонів платежів | Користувач повинен мати рахунок | Користувач може перевести гроші за допомогою предзаповнених шаблонів |

1.4.1 Розроблення функціональних вимог

Залежності між функціональними вимогами зображенні на рисунку 1.5.



Рисунок 1.5 - Діаграма функціональних вимог

Таблиця 1.3 – Опис функціональних вимог

| UC ID | FR ID | Назва | Опис |
|-------|---------|------------------------------------|---|
| UC1.0 | FR1.0.0 | Вхід в додаток | Система повинна надати сторінку для входу |
| UC1.1 | FR1.1.0 | Вхід через імейл та пароль | Система повинна надати сторінку із полями входу “Імейл” та “Пароль” та кнопкою “Увійти” |
| UC1.1 | FR1.1.1 | Вхід через імейл та пароль помилка | Якщо користувач ввів некоректні дані вивести помилку “Імейл чи пароль некоректний!” |
| UC1.2 | FR1.2.0 | Вхід через OTP | Система повинна надати користувачу поле “Номер телефону” та кнопку “Відправити СМС”, після система повинна надати поле “Пін код” та кнопку “Відправити” |
| UC1.2 | FR1.2.1 | Вхід через OTP помилка | Якщо користувач ввів некоректний дані вивести помилку “Пін код некоректний!” |
| UC2.0 | FR2.0.0 | Реєстрація в додатку | Система повинна надати користувачу поля “Імейл”, “Пароль” та кнопку “Зареєструватися”. |

Продовження таблиці 1.3

| | | | |
|-------|---------|---------------------------------------|--|
| UC3.0 | FR3.0.0 | Редагування користувачької інформації | Система повинна надати користувачу поля “Имейл”, “Пароль”, “ПІБ”, “Моб. телефон” та кнопку “Змінити”. |
| UC4.0 | FR4.0.0 | Управління рахунками | Система повинна надати користувачу сторінку із функціоналом для управління рахунками. |
| UC4.1 | FR4.1.0 | Додавання банківських карток | Система повинна надати сторінку із полями “Номер картки”, “CVV”, “Дата закінчення”, та кнопку “Додати” |
| UC4.2 | FR4.2.0 | Створення карток | Система повинна надати сторінку із списком доступних карт та кнопкою “Замовити” |
| UC4.3 | FR4.3.0 | Видалення банківських карток | Система повинна надати кнопку “Видалети” та функціонал по видаленню карток із системи. |
| UC4.4 | FR4.4.0 | Видалення рахунку | Система повинна надати кнопку “Видалети” та функціонал по видаленню рахунка користувача. |

Продовження таблиці 1.3

| | | | |
|-------|---------|-----------------------------|---|
| UC4.5 | FR4.5.0 | Перегляд балансу | Система повинна надати сторінку де буде відображений стан активів по кожній валюті користувача |
| UC4.6 | FR4.6.0 | Перегляд історію транзакцій | Система повинна надати сторінку де буде відображений історію транзакцій користувача |
| UC4.7 | FR4.7.0 | Додавання крипто-рахунку | Система повинна надати сторінку із полями “Приватний ключ”, “Публічний ключ”, кнопку “Додати” |
| UC4.8 | FR4.8.0 | Створення крипто-рахунку | Поле для вибору криптовалюти та кнопка “Додати” |
| UC4.9 | FR4.9.0 | Видалення крипто-рахунку | Система повинна надати кнопку “Видалети” та функціонал по видаленню крипто-рахунку користувача. |
| UC5.0 | FR5.0.0 | Переведення коштів | Система повинна надати користувачу форму переведення коштів |
| UC5.1 | FR5.1.0 | Переведення між картками | Система повинна надати користувачу поля “Відправник” та “Отримувач” та кнопку “Відправити” |

Продовження таблиці 1.3

| | | | |
|-------|---------|--|--|
| UC5.2 | FR5.2.0 | Переведення між рахунками | Система повинна надати користувачу поля “Відправник” та “Отримувач” та кнопку “Відправити” |
| UC5.3 | FR5.3.0 | Переведення між крипто-рахунку та фіатними рахуном | Система повинна надати користувачу поля “Відправник” та “Отримувач” та кнопку “Відправити” |
| UC5.4 | FR5.4.0 | Переведення між крипто-рахунками | Система повинна надати користувачу поля “Відправник” та “Отримувач” та кнопку “Відправити” |
| UC6.0 | FR6.0.0 | Використання шаблонів платежів | Система повинна надати предзаповнені шаблони на платіж. |

В таблиці 1.4 зображено матрицю трасування основних варіантів використання де продемонстрована залежність між користувацькими вимогами та функціональними вимогами.

Таблиця 1.4 - Матриця трасування основних варіантів використання

| | UC1.0 | UC1.1 | UC1.2 | UC2.0 | UC3.0 | UC4.0 | UC4.1 | UC4.2 | UC4.3 | UC4.4 | UC4.5 | UC4.6 | UC4.7 | UC4.8 | UC4.9 | UC5.0 | UC5.1 | UC5.2 | UC5.3 | UC5.4 | UC6.0 |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| FR1.0.0 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| FR1.1.0 | | x | | | | | | | | | | | | | | | | | | | |
| FR1.1.1 | | x | | | | | | | | | | | | | | | | | | | |
| FR1.2.0 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| FR1.2.1 | | | x | | | | | | | | | | | | | | | | | | |
| FR2.0.0 | | | | x | | | | | | | | | | | | | | | | | |
| FR3.0.0 | | | | | x | | | | | | | | | | | | | | | | |
| FR4.0.0 | | | | | | x | | | | | | | | | | | | | | | |
| FR4.1.0 | | | | | | | x | | | | | | | | | | | | | | |
| FR4.2.0 | | | | | | | | x | | | | | | | | | | | | | |
| FR4.3.0 | | | | | | | | | x | | | | | | | | | | | | |
| FR4.4.0 | | | | | | | | | | x | | | | | | | | | | | |
| FR4.5.0 | | | | | | | | | | | x | | | | | | | | | | |
| FR4.6.0 | | | | | | | | | | | | x | | | | | | | | | |
| FR4.7.0 | | | | | | | | | | | | | x | | | | | | | | |
| FR4.8.0 | | | | | | | | | | | | | | x | | | | | | | |
| FR4.9.0 | | | | | | | | | | | | | | | x | | | | | | |
| FR5.0.0 | | | | | | | | | | | | | | | | x | | | | | |
| FR5.1.0 | | | | | | | | | | | | | | | | | x | | | | |
| FR5.2.0 | | | | | | | | | | | | | | | | | | x | | | |
| FR5.3.0 | | | | | | | | | | | | | | | | | | | x | | |
| FR5.4.0 | | | | | | | | | | | | | | | | | | | | x | |
| FR6.0.0 | | | | | | | | | | | | | | | | | | | | | x |

В таблиці 1.5 зображено матрицю основних залежності між функціональними вимогами де продемонстрована залежність функціональними вимогами відносно функціональними вимогами.

Таблиця 1.5 - Матриця основних залежностей між функціональними

| | FR1.0.0 | FR1.1.0 | FR1.1.1 | FR1.2.0 | FR1.2.1 | FR2.0.0 | FR3.0.0 | FR4.0.0 | FR4.1.0 | FR4.2.0 | FR4.3.0 | FR4.4.0 | FR4.5.0 | FR4.6.0 | FR4.7.0 | FR4.8.0 | FR4.9.0 | FR5.0.0 | FR5.1.0 | FR5.2.0 | FR5.3.0 | FR5.4.0 | FR6.0.0 |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| FR1.0.0 | x | | | | | | | | | | | | | | | | | | | | | | |
| FR1.1.0 | x | x | | | | | | | | | | | | | | | | | | | | | |
| FR1.1.1 | x | | x | | | | | | | | | | | | | | | | | | | | |
| FR1.2.0 | x | | | x | | | | | | | | | | | | | | | | | | | |
| FR1.2.1 | x | | | | x | | | | | | | | | | | | | | | | | | |
| FR2.0.0 | | | | | | x | | | | | | | | | | | | | | | | | |
| FR3.0.0 | | | | | | | x | | | | | | | | | | | | | | | | |
| FR4.0.0 | | | | | | | | x | | | | | | | | | | | | | | | |
| FR4.1.0 | | | | | | | | x | x | | | | | | | | | | | | | | |
| FR4.2.0 | | | | | | | | x | | x | | | | | | | | | | | | | |
| FR4.3.0 | | | | | | | | x | | | x | | | | | | | | | | | | |
| FR4.4.0 | | | | | | | | x | | | | x | | | | | | | | | | | |
| FR4.5.0 | | | | | | | | x | | | | | x | | | | | | | | | | |
| FR4.6.0 | | | | | | | | x | | | | | | x | | | | | | | | | |
| FR4.7.0 | | | | | | | | x | | | | | | | x | | | | | | | | |
| FR4.8.0 | | | | | | | | x | | | | | | | | x | | | | | | | |
| FR4.9.0 | | | | | | | | x | | | | | | | | | x | | | | | | |
| FR5.0.0 | | | | | | | | | | | | | | | | | | x | | | | | |
| FR5.1.0 | | | | | | | | | | | | | | | | | | x | x | | | | |
| FR5.2.0 | | | | | | | | | | | | | | | | | | x | | x | | | |
| FR5.3.0 | | | | | | | | | | | | | | | | | | x | | | x | | |
| FR5.4.0 | | | | | | | | | | | | | | | | | | x | | | | x | |
| FR6.0.0 | | | | | | | | | | | | | | | | | | | | | | | x |

1.4.2 Розроблення нефункціональних вимог

Додаток повинен підтримувати наступні браузері:

- Chrome – latest;
- Firefox – latest;
- Edge - 2 most recent major versions;
- IE – 11;
- IE – 10;
- IE – 9;
- IE Mobile – 11;
- Safari - 2 most recent major versions;
- iOS - 2 most recent major versions;
- Android - Nougat (7.0);
- Android - Marshmallow (6.0);
- Android - Lollipop (5.0, 5.1);
- Android - KitKat (4.4).

Додаток повинен підтримувати наступні мови:

- англійська.

Додаток повинен мати адаптивну верстку сайту. Особливістю адаптивного сайту є те, що його макет і вся верстка змінюється залежно від розміру екрана на якому він відображається. Передбачена підтримка наступних розмірів екранів:

- ПК-Стандартний;
- мобільний;
- планшетний;
- ПК-Широкоформатний.
- від 320px (мінімальний дозвіл) і до 1920px (Full HD).

Додаток повинен буди доступний для людей із інвалідності.

Також, розробка веб застосунку має декілька основних етапів. Такі як розробка інтерфейсу - макету цільового продукту та проектування архітектури. Перший етап - це розробка інтерфейсу побудованих на принципах UX та UI [2]. Інтерфейс застосування повинен слідувати наступним правилам:

- бути простий в освоєнні;
- мати інтуїтивно зрозумілу структуру;
- бути інформативним;
- не бути переповненим надлишковою інформацією;
- мати відповідність очікувань із результатом при взаємодії із ресурсом;
- мати баланс кольорів;
- не бути переповненим надмірним цвітовим наповненням;
- використовувати очевидні цвітові вказівники;
- дотримуватися єдиного патерну при розробці інтерфейсу;
- мати приємний інтерфейс.

Також інтерфейс повинен бути побудована на загальних принципах розміщення ключових елементів інтерфейсу. Наприклад, меню повинно розміщуватися зверху або збоку екрану, не міняти свого положення на всіх сторінках сайту, та мати відповідне наповнення. Для зручності максимально рекомендована кількість пунктів меню дорівнює семи.

Інтерфейс повинен доступним та надавати швидкий доступ до найбільш активних(важливих) елементі, шлях користувача для виконання певних пріоритетних дій повинен бути мінімізований. Найбільш важливі елементи інтерфейсу повинні бути на видному та легко доступному місті, а рідко-змінні елементи повинні не засмічувати інтерфейс, наприклад, зміна паролю - елемент з яким найменш часто взаємодіють, повинен розмінюватися в розділі налаштування, безпека, а переказ на рахунок, навпаки, повинен бути на основній сторінці - оскільки це найважливіший елемент для користувача.

Другий етап - це розробка архітектури. Для ефективної та швидкої реалізації необхідно мати зручний технічний інструмент. Тому даний застосунок буде мати такі компоненти.

Фейкові API. Для опрацювання даних повинен бути гнучкий інструмента за для імітації роботи із базою даних, швидкий в роботі, простий в відслідковувати помилок.

Pop-up чи спливаючи вікна. Для роботи із користувачем повинна бути система вікон підтвердження та повідомлень.

Angular Таблиці. Для відображення великого об'єму даних повинна бути таблиця з будованою фільтрацією по стовпцям та присутністю пагінацією.

Бібліотеки для роботи із CSS та HTML.

Також, для регуляції доступного для користувача функціоналу повинна бути регуляція по рівням доступу.

1.4.3 Постановка комплексу завдань модулю

Метою даного є вдосконалення методів керування коштами для приватної особи, за допомогою WEB-застосунку.

Призначення застосунку це надання нового функціоналу для користувача за для покращення процесів переведення, обміну, купівлі та продажу коштів.

Завданням до даної роботи є:

- створення інструментів для фізичних осіб, по управління крипто та фіатними коштами;
- створення інструменту для динамічного відображення повідомлень pop-up;
- створення інструменту для обробки, імітації серверної частини - mock-api;
- створення інструменту для опрацювання даних в розширеному табличному виді - angular table.

Застосунок повинен мати адаптивну верстку сайту та підтримувати останні версії вказані браузерів.

1.5 Висновки по розділу

В даному розділі був проведено аналіз вимог до програмного забезпечення, а саме було описано загальні базиси структури та підходи. Було проаналізовано та виділено основний набір сервісів.

Було проведено аналіз популярних рішень, таких як Приват24 та Сави, і виділені ключові моменти технічних рішень, проведено аналіз архітектурних рішень та основних модулів для розробки (pop-up, mock-api, table). Та рішення відомих програмних продуктів UXDA.

Проаналізовано та виділено основні функції та користувацькі вимоги. Будо зображено Use-case діаграма користувацьких вимог. Та розписано функціональні вимоги до кожного користувацького кейсу та зображено від повідні діаграми:

- діаграма функціональних вимог;
- діаграма залежності між функціональних вимог;
- матриця трасування основних варіантів використання.

Було описано нефункціональних вимог, вимог до браузеру, мови інтерфейсу та розмірів екрану та доступність.

Після проведеного аналізу було поставно комплекс завдань до застосунку інтерфейсу та архітектури.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

При розробці програмного забезпечення необхідно провести детальне моделювання основних процесів та структуризації архітектури для майбутнього застосунку. Для належного моделювання процесів було створено діаграму BPMN. Схема процесів діяльності представлена на рисунку 2.1.

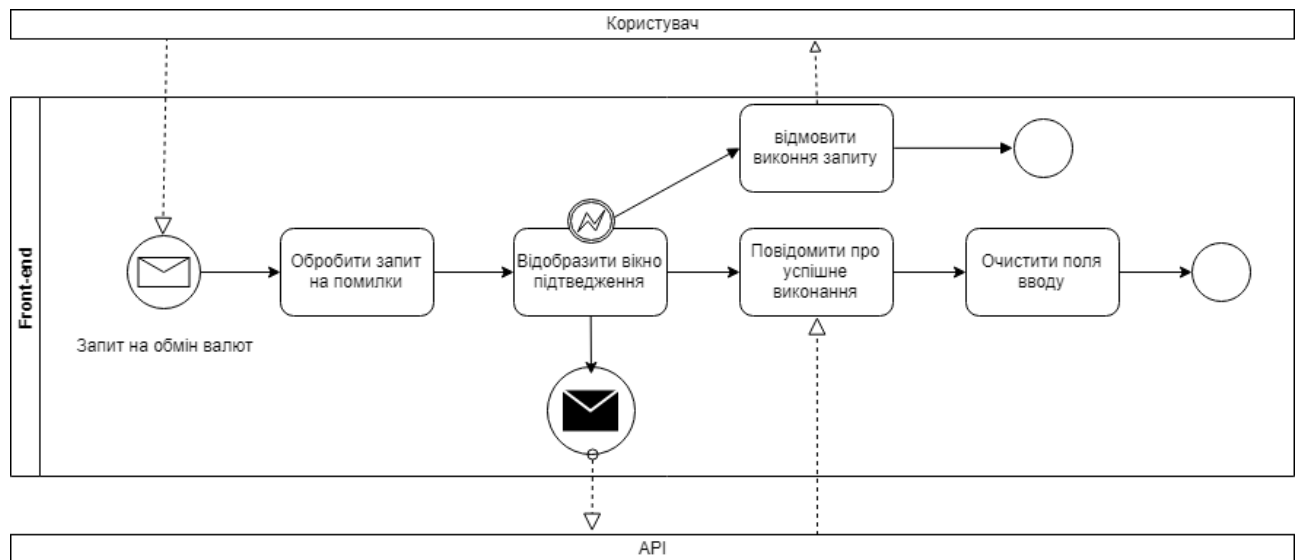


Рисунок 2.1 - Схема підтвердження переказу

На діаграмі підтвердження переказу наступний опис:

- користувач надсилає дані для переказу коштів;
- система обробляє запит на наявність помилок;
- система відображає вікно підтвердження для користувача
- якщо користувач відмовляє у переведенні то система закінчує роботу;
- система повідомляє користувача про виконання переведення;
- система очищує поля вводу користувача.

Нижче представлена діаграма станів для переведення коштів з одного рахунку на інший (рисунок 2.1).

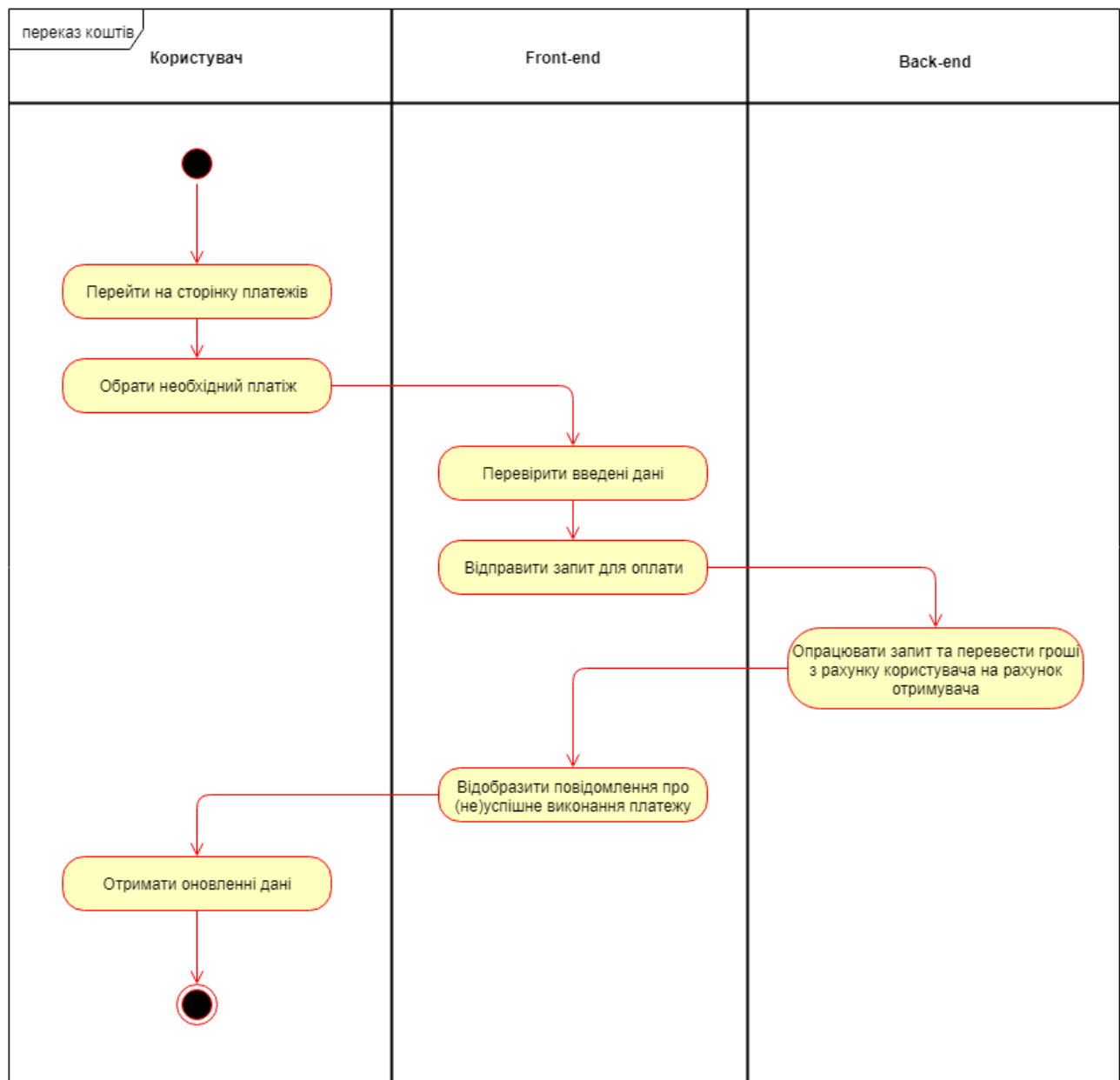


Рисунок 2.3 - Переказ коштів

Послідовний опис процесу переказу коштів:

- перейти на сторінку платежів;
- обрати необхідний платіж;
- перевірити введені дані;
- відправити запит для оплати;

- опрацювати запит та перевести гроші з рахунку користувача на рахунок отримувача;
- відобразити повідомлення про (не)успішне виконання платежу;
- отримати оновленні дані.

Діаграма проведення операції по обміну грошей різних валют (рисунок 2.4).

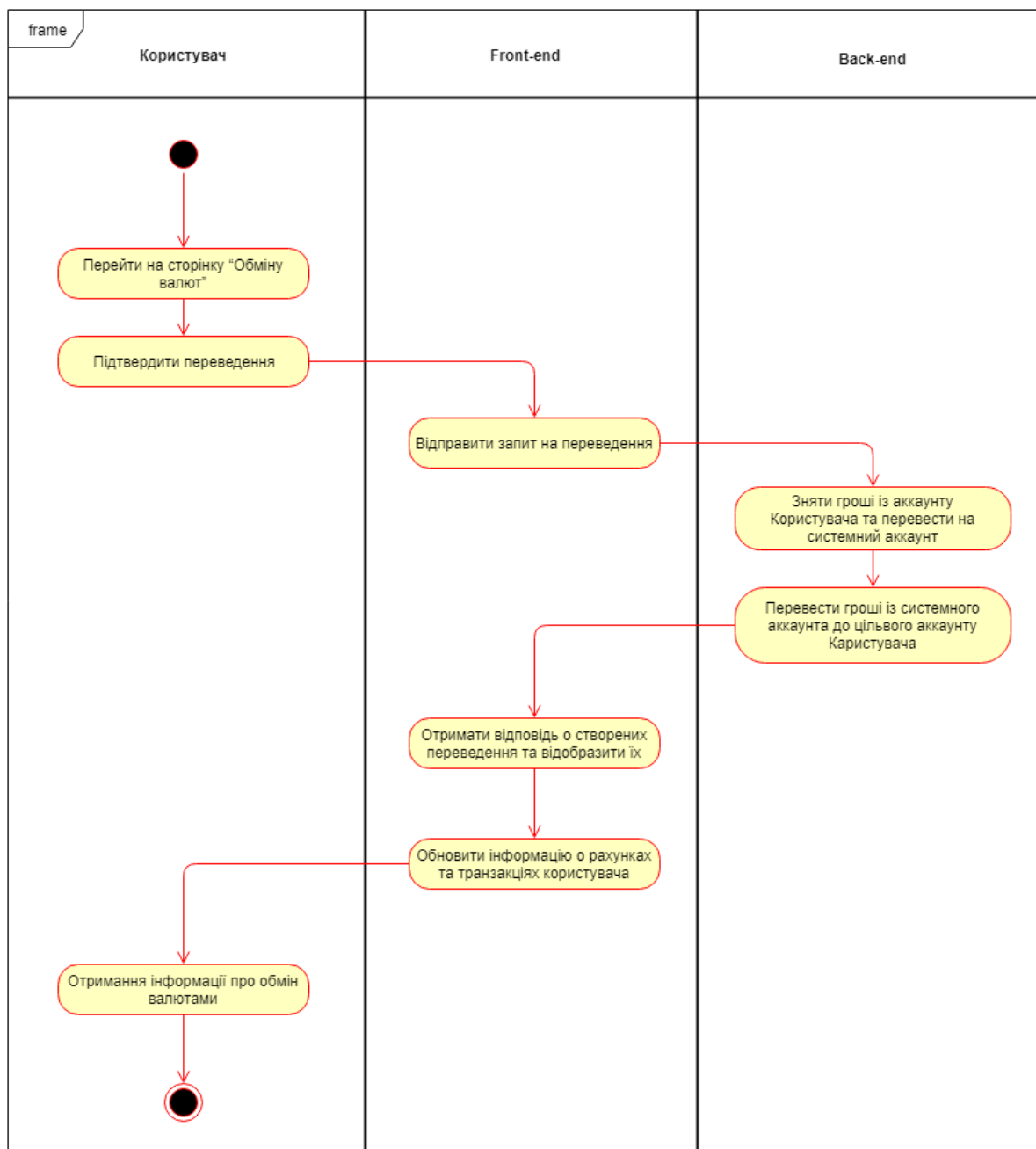


Рисунок 2.4 – Купівля-продаж валюти

Послідовний опис процесу переказу купівля-продаж валюти:

- перейти на сторінку “Обміну валют”;
- підтвердити переведення;
- відправити запит на переведення;
- зняти гроші із аккаунту(облікового запису) користувача та перевести на системний аккаунт;
- перевести гроші із системного аккаунта до цільового аккаунту Користувача;
- отримати відповідь о створених переведення та відобразити їх;
- оновити інформацію о рахунках та транзакціях користувача;
- отримання інформації про обмін валютами.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 40 |

2.2 Архітектура програмного забезпечення

Для написання програмного забезпечення було обрано загальнодоступний відкритий фреймовк Angular[1], а саме останню його версію 7.2.15.

Angular - це платформа і основа для побудови клієнтських додатків у HTML і TypeScript. Основними будівельними блоками програми Angular є Модулі, які забезпечують блоки компіляції для компонентів. Модулі збирають відповідний код у функціональні групи;

Додаток має кореневий модуль App, який дозволяє завантажувати початкову програму, і багато інших модулів.

Компоненти використовують сервіси, які забезпечують певну функціональність, для роботи із даними та не пов'язану з інтерфейсом. Сервіси можуть бути введені в компоненти як залежності, що робить ваш код модульним, багаторазовим і ефективним.

Як компоненти, так і сервіси - це просто класи, де декоратори позначають їх тип і надають метадані, які повідомляють Angular, як їх використовувати.

Метадані для класу компонентів асоціюють його з шаблоном, який визначає подання. Поєднують звичайний HTML з директивами Angular та розміткою, що дозволяє Angular змінити HTML перед його відображенням.

Компоненти програми зазвичай багат шарові та організовані ієрархічно. Angular надає послугу маршрутизатора, який допомагає вам визначити шляхи навігації серед переглядів. Маршрутизатор забезпечує складні навігаційні можливості браузера.

Структура проекту та ієрархія рисунок 2.5 файлової система рисунок 7.2 була побудована за допомогою рекомендацій від розробників фреймову angular. Вхідним елементом в застосунок слугує компонент App.

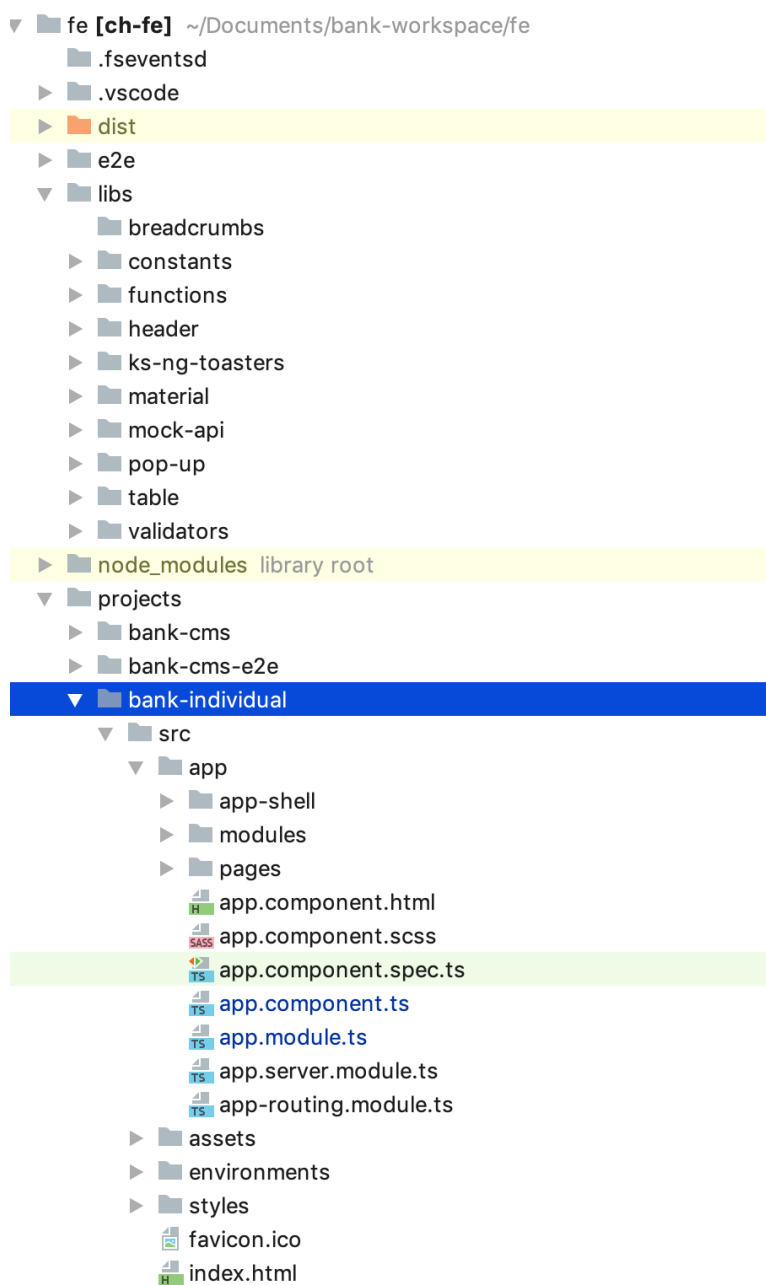


Рисунок 2.5 - Файлова схема

Схема архітектури описує основну структура та включає головні компоненти для створення коректного рішення.

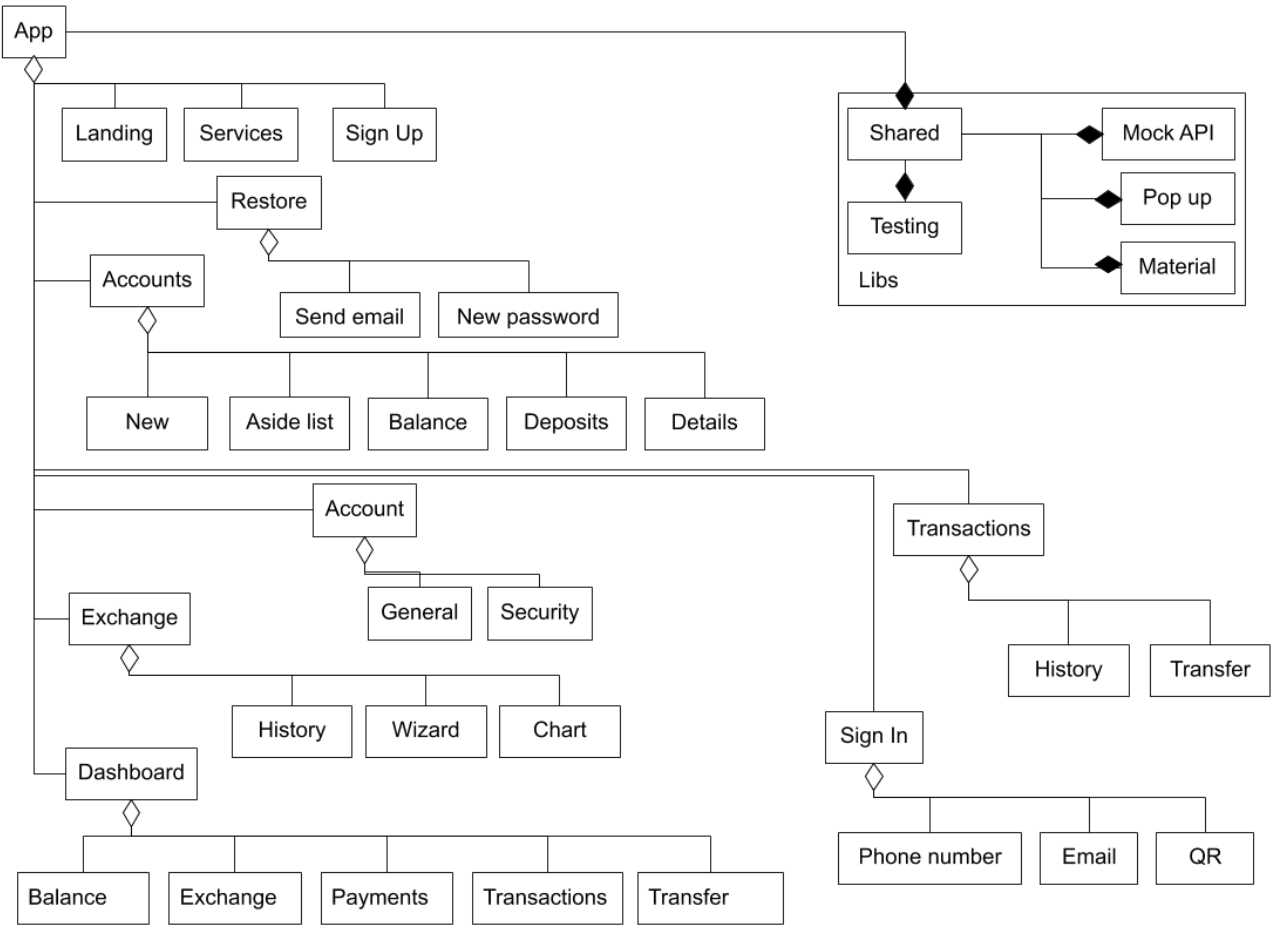


Рисунок 2.5 - Схема зав'язків архітектури компонентів

Опис фейкового API:

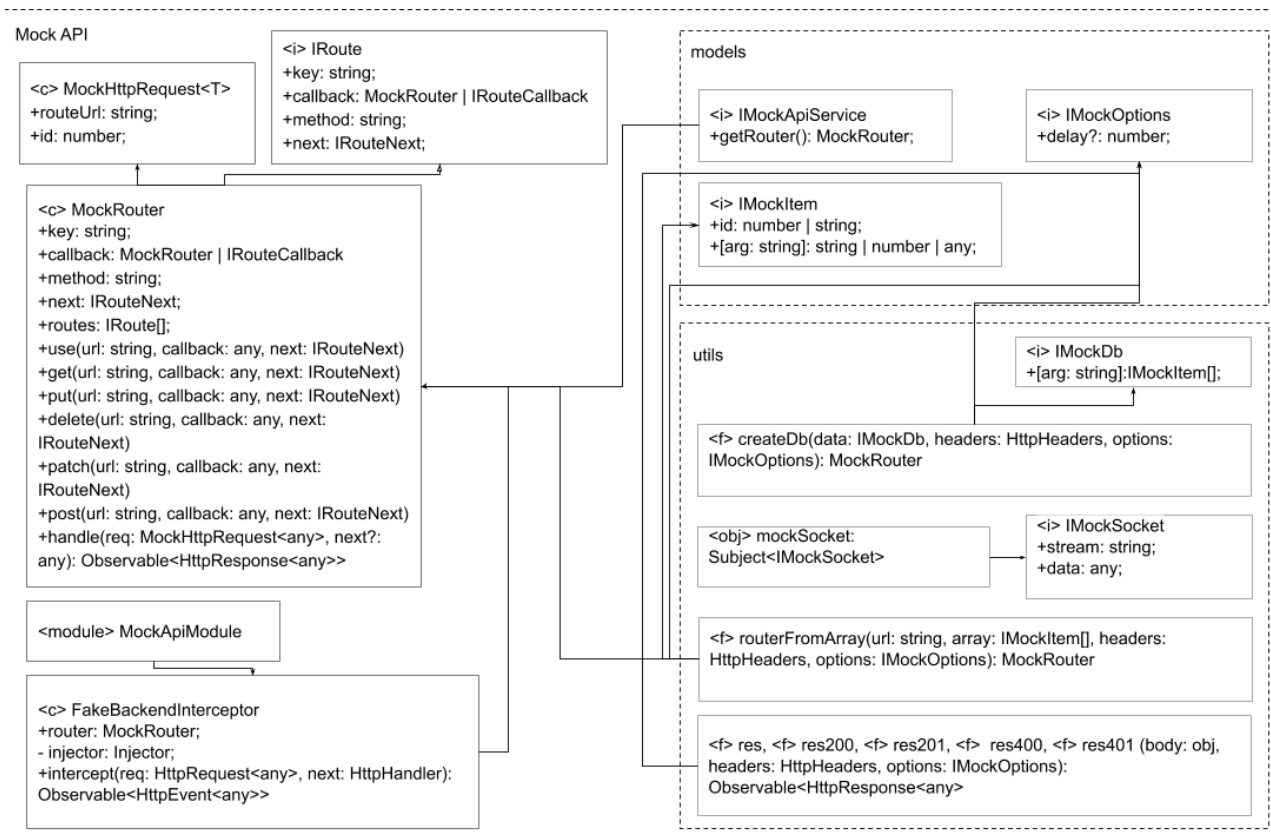


Рисунок 2.6 - UML діаграма фейк API

Клас MockRouter реалізований патерном композит описує маршрутизатор для рест api:

- поле key - адреса запиту;
- поле callback - функція виклику у разі співпадіння з запитом;
- поле method - тип запиту;
- поле routes - масив маршрутизаторів;
- поле next - наступний роутер у разі пропуску;
- метод use - розширяє масив обробником різного типу;
- метод get - обробник запиту GET;
- метод put - обробник запиту PUT;
- метод delete - обробник запиту DELETE;

- метод patch - обробник запиту PATCH;
- метод post - обробник запиту POST;
- метод handle - оброблює запит користувача.

Функція routerFromArray - генерує роутер із масиву.

Функція res, функція res200, функція res201, функція res400, функція res401 - генерують Http відповідь.

Функція createDb - функція породжувач, генерує базу даних і рест структуру із об'єктів чи json.

Об'єкт mockSocket - створює симуляцію сокетів.

Клас FakeBackendInterceptor - сервіс необхідний для обробки запитів:

- поле router - основний маршрутизатор;
- функція intercept - слугує прокси для запитів;
- поле injector - надає доступ до сервісів застосунку.

Модуль MockApiModule - груперує залежності та описує конфігурації.

Опис Pop-up:

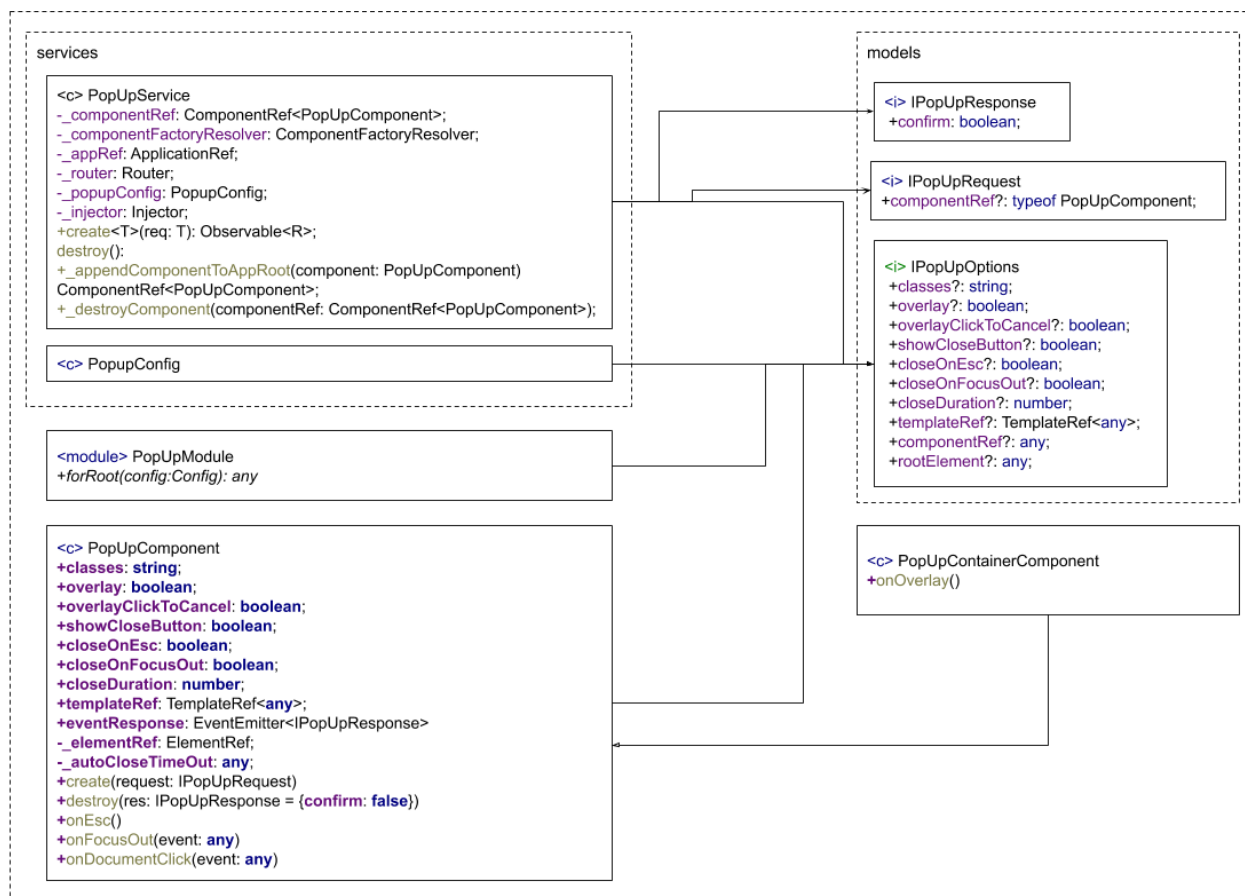


Рисунок 2.7 - UML діаграма Pop-up

Модуль `PopUpModule` - групує залежності та описує конфігурації.

Компонент `PopUpService` - реалізує взаємодію через сервіс:

- поле `_componentRef` - посилання на конкретний компонент;
- поле `_componentFactoryResolver` - фабрика для створення компонентів;
- поле `_appRef` - посилання на рутовий елемент;
- поле `_router` - сервіс маршрутизації;
- поле `_popupConfig` - конфігурація;
- поле `_injector` - надає доступ до сервісів застосунку;
- функція `create` - створення;
- функція `destroy` - видалення;

PopupConfig

- поле classes - класи;
- поле overlay - демонструвати фон;
- поле overlayClickToCancel - перевірка на закриття через фон;
- поле showCloseButton - перевірка на відображення через кнопку

Закрити;

- поле closeOnEsc - перевірка на закриття через кнопку Esc;
- поле closeOnFocusOut - перевірка на закриття через залишення

фокусу;

- поле closeDuration - перевірка на закриття через проміжок часу;
- поле templateRef - динамічне наповнення;
- поле componentRef - посилання на конкретний компонент;
- поле rootElement - компонент;

Інтерфейс IPopUpRequest - опис запиту.

Інтерфейс IPopUpOptions - опис конфігурації.

Інтерфейс - IPopUpResponse - опис відповіді;

Об'єкт - DEFAULT_POPUP_CONFIG - налаштування за замовчуванням.

Компонент PopUpComponent - основна реалізація поведінки поп-апа:

- поле classes - класи;
- поле overlay - демонструвати фон;
- поле overlayClickToCancel - перевірка на закриття через фон;
- поле showCloseButton - перевірка на відображення через кнопку

Закрити;

- поле closeOnEsc - перевірка на закриття через кнопку Esc;
- поле closeOnFocusOut - перевірка на закриття через залишення

фокусу;

- поле closeDuration - перевірка на закриття через проміжок часу;
- поле templateRef - динамічне наповнення;
- поле eventResponse - відповідь при закритті;

- поле `_elementRef` - посилання на едемент;
- поле `_autoCloseTimeOut` - час автоматичного закриття;
- метод `create` - створити;
- метод `destroy` - знищити компонент;
- метод `onEsc` - обробка натиснення Esc;
- метод `onFocusOut` - обробка загублення фокусу в попапу;
- метод `onDocumentClick` - обробка загублення фокусу над попапом.

Компонент `PopUpContainerComponent` - огортка над попапом для динамічного заповнення:

- метод `onOverlay` - оброблювання натиску на фон.

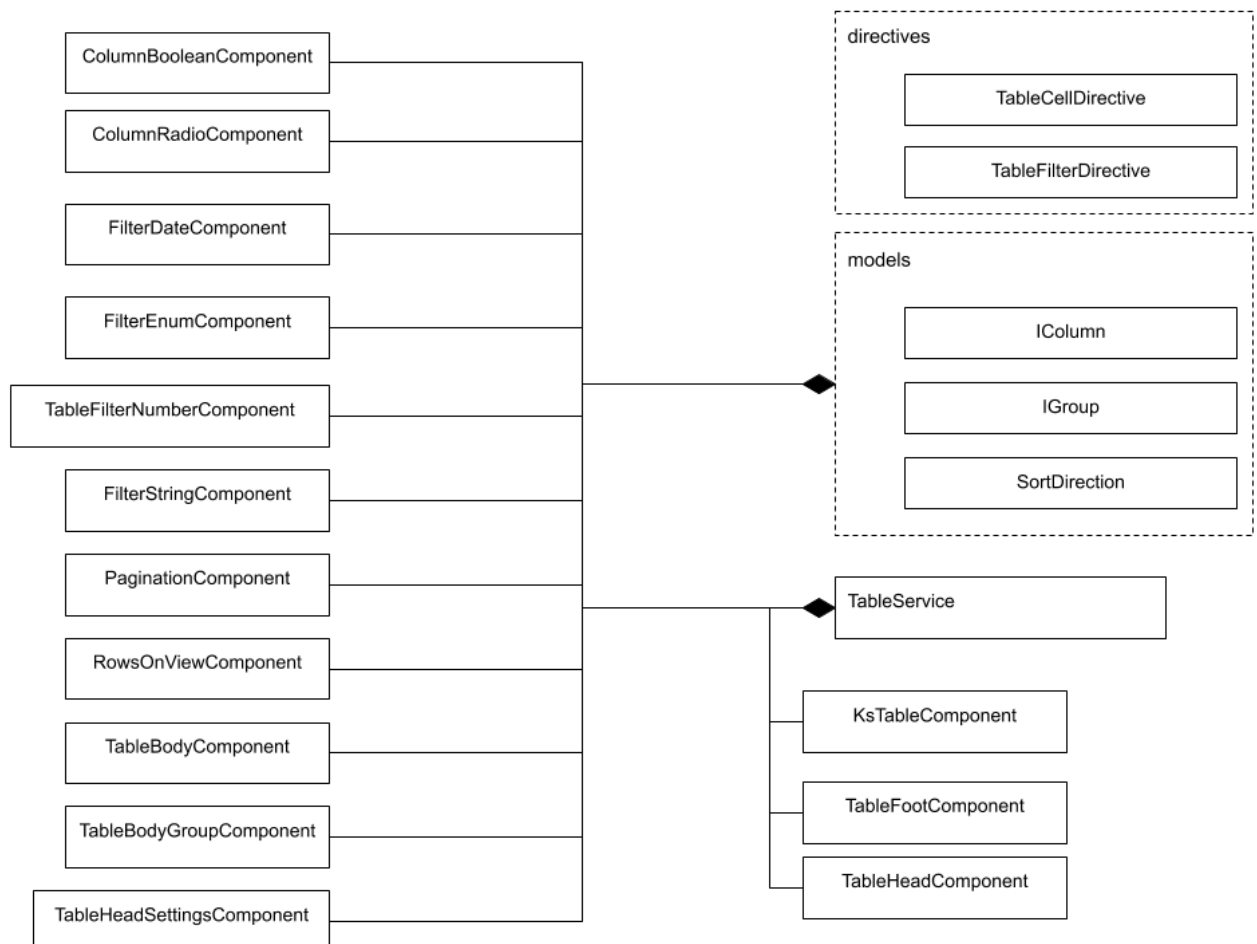


Рисунок 2.5 - Схема зв'язків Angular Table

2.3 Конструювання програмного забезпечення

Основою для збереження та опрацювання даних слугує структура даних JSON. Для розробки даного застосунку використовується Фейк API в основі якого лежить структура даних (рисунок 2.6) та реалізація RESTful архітектури для комунікації між інтерфейсом та даними.

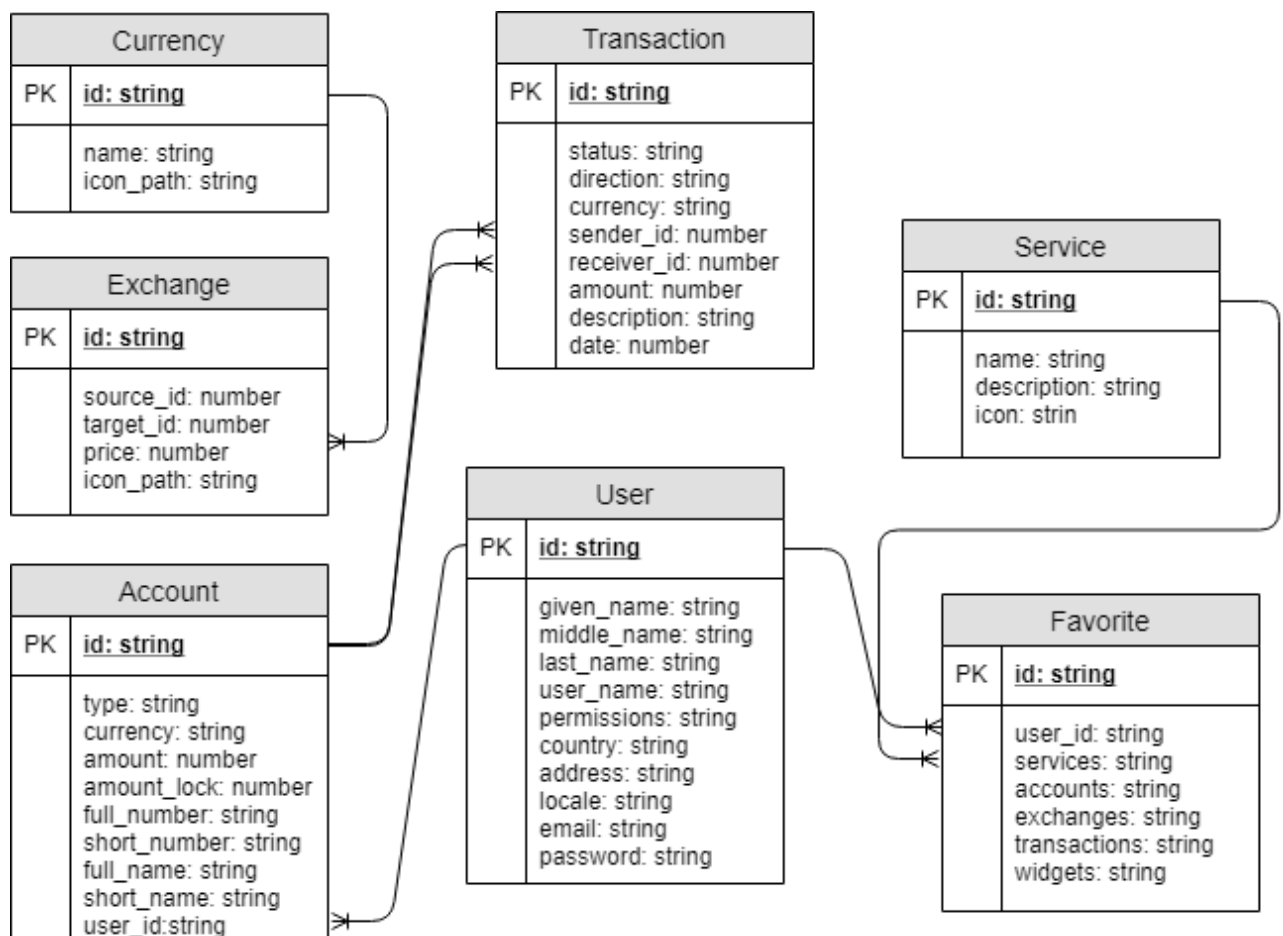


Рисунок 2.6 – Структура даних JSON

Об'єкт Service реалізує збереження наступних полів:

- id - унікальний ідифікатор;
- name - назва;
- description - опис транзакції;
- icon - логотип компанії.

Об'єкт User реалізує збереження наступних полів:

- id - унікальний ідифікатор;
- given_name - ім'я;
- middle_name - фамілія;
- user name - ім'я користувача;
- permissions - рівень доступу;
- country - країна;
- address - адреса;
- locale - локаль;
- email - имейл;
- password - пароль.

Об'єкт Exchange реалізує збереження наступних полів:

- id - унікальний ідифікатор;
- source - ресурс валюти;
- target - цільова валюта;
- icon path - іконка валюти.

Об'єкт Account реалізує збереження наступних полів:

- id - унікальний ідифікатор;
- type - тип рахунку;
- currency - валюта рахунку;
- amount - сума на рахунку;
- amount Lock - сума заблоковано на рахунку;
- full number - номер рахунку;
- short number - короткий рахунок;
- full Name - повна назва;
- short name - коротка назва.

Об'єкт Transaction реалізує збереження наступних полів:

- id - унікальний ідифікатор;
- status - статус транзакції;

- direction - напрям транзакції;
- currency - валюта;
- sender_id - вказівник на відправника;
- receiver_id - вказівник на отримувача;
- amount - сума транзакції;
- description - опис транзакції;
- date - дата транзакції.

2.4 Аналіз безпеки даних

Для захисту даних, було використано кращі практики написання програмного коду та рекомендації від розробників angular[1].

Застосунок було розроблено за допомогою фреймову Angular, а в ньому закладені певні захисні системи. Найважливіша з них – захист від атак типу XSS.

XSS (англ. Cross Site Scripting) — тип вразливості інтерактивних інформаційних систем. XSS виникає, коли на сторінки, які були згенеровані сервером, потрапляють користувацькі js - скрипти. Специфіка подібних атак полягає в тому, що замість безпосередньої атаки сервера зловмисники використовують вразливий сервер для атаки на користувача.

Довгий час програмісти не приділяли їм належної уваги, вважаючи їх безпечними. Однак ця думка помилкова: на сторінці або в HTTP-Cookie можуть бути досить вразливі дані (наприклад, ідентифікатор сесії адміністратора). На популярному сайті скрипт може влаштувати DoS-атаку.

Використовуючи Angular, ми захищені від даного типу атак за замовчуванням. Angular систематично блокує будь-які XSS помилки, він приймає всі значення, як підозрілі за замовчуванням. Коли значення додається в DOM, через шаблон, властивість, атрибут, стиль або інтерполяцію. Він видаляє та уникає всі підозрілі значення.

2.5 Висновки по розділу

Було розроблено один з найважливіших розділів це конструювання, розробка документації з опис ключових модулів додатку, описана структура бази даних та деталізовані моделі.

В даному розділі були описані всі технічні і програмні засоби, що використовувалися для створення системи. Описані технології, завдяки яким була розроблена система. Було описано діаграми станів для проведення платежів та купівлі валюти.

Завданням розробки було створити, застосунок за наступними правилами:

- гнучкий у масштабуванні;
- легкий в інтегруванні в різних системах;
- унітарність функціональних модулів;
- перевикористовування та абстракція функціоналу .

Була описана архітектура додатку, структура файлової системи та розміщення компонентів, наведені діаграми. Проведено аналіз безпеки даних.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Процес тестування програмного забезпечення, являється важливим етапом життєвого циклу розробки. Цей процес дозволяє не лише перевірити якість готового продукту, а й перевірити виконання, поставлених вимог та функцій, які мають бути реалізовані. З допомогою тестування, можна охопити різні елементи застосунку, опираючись на наявні часові та людські ресурси. Та виявити критичні проблеми, до здійснення впровадження.

Для тестування коду використовується вбудований механізм юніт тестів та інтеграційних тестів.

Тестуванню підлягає наступний функціонал:

- належне відображення елементів інтерфейсу;
- належна поведінка елементів після виконня дій користувача;
- адаптованість застосунку;
- швидкість роботи застосунку.

3.2 Опис процесів тестування

Для тестування створеного застосунку було обрано метод функціонального тестування, за допомогою якого можна перевірити правильність виконання поставлених задач.

Було вирішено перевірити наступні основні функції застосунку:

- реєстрація у застосунку (Таблиця 3.1);
- авторизація у застосунку (Таблиця 3.2);
- аутентифікації у застосунку ОТР(Таблиця 3.3);
- перевірка правильності роботи кнопки «Вихід» (Таблиця 3.4);
- відеозображення рахунків користувача (Таблиця 3.5);
- переказ між рахунками (Таблиця 3.6);

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 53 |

- перевірка правильності роботи додавання картки банку (Таблиця 3.7);
- редагування користувацької інформації (Таблиця 3.8).

Залежність між функціональними вимогами та тестами зображена на рисунку 3.1.

| | реєстрації у застосунку. | аутентифікації у застосунку. | аутентифікації у застосунку ОТР | роботи кнопки «Вихід». | роботи сторінки рахунків. | роботи переказу коштів. | роботи додавання картки банку. | редагування користувацької інформації. |
|---------|--------------------------|------------------------------|---------------------------------|------------------------|---------------------------|-------------------------|--------------------------------|--|
| FR1.0.0 | | x | x | x | | | | |
| FR1.1.0 | | x | | | | | | |
| FR1.1.1 | | x | | | | | | |
| FR1.2.0 | | | x | | | | | |
| FR1.2.1 | | | x | | | | | |
| FR2.0.0 | x | | | | | | | |
| FR3.0.0 | | | | | | | | x |
| FR4.0.0 | | | | | x | | x | |
| FR4.1.0 | | | | | | | x | |
| FR4.2.0 | | | | | | | | |
| FR4.3.0 | | | | | | | | |
| FR4.4.0 | | | | | | | | |
| FR4.5.0 | | | | | x | | | |
| FR4.6.0 | | | | | x | | | |
| FR4.7.0 | | | | | | | | |
| FR4.8.0 | | | | | | | | |
| FR4.9.0 | | | | | | | | |
| FR5.0.0 | | | | | | x | | |
| FR5.1.0 | | | | | | x | | |

Рисунок 3.1 - Залежність функціональних вимог та тестів

3.3 Опис контрольного прикладу

Таблиця 3.1 – Перевірка можливості реєстрації у застосунку

| | |
|--|---|
| Мета тесту | Перевірка можливості реєстрації у застосунку. |
| Початковий стан | Відкритий додаток. |
| Вхідні дані | Пароль та поштова скринька користувача |
| Схема проведення тесту | Відкрити додаток. На відображеній головній сторінці натиснути на кнопку «Реєстрація»; У відкритому вікні ввести дані для реєстрації: email, та password, натиснути кнопку «Реєстрація». |
| Очікуваний результат | Застосунок успішно реєструє користувача, створює профіль на основі наданої інформації відкриває головну сторінку додатку. |
| Стан програмного продукту після проведення випробувань | Зареєстрований новий користувач, система створює новий профіль з надано інформацією. |

Таблиця 3.2 – Перевірка можливості аутентифікації у застосунку

| | |
|------------------------|---|
| Мета тесту | Перевірка можливості аутентифікації у застосунку. |
| Початковий стан | Відкритий додаток. |
| Вхідні дані | Логін, пароль користувача |
| Схема проведення тесту | Відкрити додаток. На відображеній головній сторінці натиснути на кнопку «Увійти»; У відкритому вікні ввести дані для аутентифікації: login та password, натиснути кнопку «Підтвердити». |

Продовження таблиці 3.2

| | |
|--|---|
| Очікуваний результат | Система успішно аутентифікація користувача та відкриває головну сторінку додатку. |
| Стан програмного продукту після проведення випробувань | Користувач Авторизований. |

Таблиця 3.3 – Перевірка можливості аутентифікації у застосунку OTP

| | |
|--|---|
| Мета тесту | Перевірка можливості аутентифікації у застосунку OTP |
| Початковий стан | Відкритий додаток. |
| Вхідні дані | Номер телефону, пінкод "4321" |
| Схема проведення тесту | Відкрити додаток. На відображеній головній сторінці натиснути на кнопку «Увійти»; У відкритому вікні ввести дані для аутентифікації: phoneNumber та password, натиснути кнопку «Підтвердити». |
| Очікуваний результат | Система успішно аутентифікація користувача та відкриває головну сторінку додатку. |
| Стан програмного продукту після проведення випробувань | Користувач Авторизований. |

Таблиця 3.4 – Перевірка правильності роботи кнопки «Вихід».

| | |
|--|--|
| Мета тесту | Перевірка правильності роботи кнопки «Вихід». |
| Початковий стан | Відкритий застосунок, користувач авторизований. |
| Вхідні дані | Відкрита сторінка “Рахунки”, кнопки «Вихід» відображаються коректно. |
| Схема проведення тесту | Натискаємо кнопку «Вихід». |
| Очікуваний результат | Користувач автоматично покидає кабінет сайту та його сесія видаляється із системи. |
| Стан програмного продукту після проведення випробувань | Відкрита головна сторінка сайту. |

Таблиця 3.5 – Відеозображення рахунків користувача.

| | |
|------------------------|--|
| Мета тесту | Перевірка правильності роботи сторінки рахунків. |
| Початковий стан | Відкритий застосунок, користувач авторизований. |
| Вхідні дані | Відкрита сторінка “Рахунки”. |
| Схема проведення тесту | Очікуємо загрузку сторінки. Перевіряльник наявність усіх рахунків користувача. Перевіряльник наявність підрахованого балансу користувача. Перевіряємо коректність підрахунків по поточному курсу оброї валюти |

Продовження таблиці 3.5

| | |
|--|--|
| Очікуваний результат | Виведені усі данні; Баланси підраховані коректно. |
| Стан програмного продукту після проведення випробувань | Відкрита сторінка “Рахунки”. |

Таблиця 3.6 – Переказ між рахунками

| | |
|--|--|
| Мета тесту | Перевірка коректності роботи переказу коштів. |
| Початковий стан | Відкритий застосунок, користувач авторизований, користувач має 2 рахунки по 100UAH на кожному. |
| Вхідні дані | Відкрита вкладка «Перекази». |
| Схема проведення тесту | Обираємо перший рахунок, обираємо 2 рахунок, вказуємо суму 100 UAH та підтверджувати переказ. |
| Очікуваний результат | Баланс на першому рахунку 0 UAH на другому 200 UAH |
| Стан програмного продукту після проведення випробувань | Відкрита вкладка «Перекази». |

Таблиця 3.7 – Перевірка правильності роботи додавання картки банку

| | |
|--|---|
| Мета тесту | Перевірка правильності роботи додавання картки банку. |
| Початковий стан | Відкритий застосунок, користувач авторизований. |
| Вхідні дані | Відкрита сторінка “Додати картку”. |
| Схема проведення тесту | Заповнити поля “Номер картки”, “Дата”, “CVV” та натиснути кнопку “Додати” |
| Очікуваний результат | Картка додана до списку карток |
| Стан програмного продукту після проведення випробувань | Повідомлення про успішне додання картки. |

Таблиця 3.8 – Перевірка правильності роботи редагування користувацької інформації

| | |
|--|---|
| Мета тесту | Перевірка правильності редагування користувацької інформації. |
| Початковий стан | Відкритий застосунок, користувач авторизований. |
| Вхідні дані | Відкрита сторінка “Налаштування”. |
| Схема проведення тесту | Заповнити поля “Імейл”, та натиснути кнопку “Змінити” |
| Очікуваний результат | Імейл змінений |
| Стан програмного продукту після проведення випробувань | Повідомлення про успішний змін імейлу. |

3.4 Висновок до розділу

В даному розділі було описано основні види та механізми тестування. Описано процедуру проведення тестів. Було протестована коректність роботи та поведінку компонентів при різних факторів впливу.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 60 |

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Рекомендовано використовувати автоматичне розгортання системи для цього достатньо зробити коміт до вже існуючого супозиторію.

Для розгортання даного застосунку вручну потрібно:

- скопіювати застосунок із репозиторію;
- виконати команду в консолі `yarn install`, попередньо встановив менеджер - проектів `yarn`;
- зібрати проект - `yarn build`;
- та скопіювати `/dist` в директорію вашого сервера;
- рекомендовано використовувати `nginx` сервер, оскільки конфігурації для нього вже розміщені в проекті;
- перезапустити сервер.

Додаткова інформація розміщена в додатку Г “Керівництво системного програміста”.

4.2 Робота з програмним забезпеченням

Робота з програмним забезпеченням описане в додатку Д “Керівництво програміста” та в додатку Є “Керівництво користувача”.

4.3 Висновок до розділу

В даному розділі було описано процедуру розгортання для застосунку в серверному отченні та описано конфігурації для налаштування і запуску додатку на сервері nginx.

Було описано Керівництво програміста в якому детально описано процедури інсталяції додатку, розгортання на локальному сервері, запуск тестів, серверну компіляцію, та норми правлення та фіксування змін

Було описано Керівництво користувача в якому було зображено усі сторінку додатку з описами процедур використання.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 62 |

ВИСНОВКИ

Було створене нове технічне рішення яке представляє собою розробку засторонку на основі веб-ресурсів для управління коштами із використанням передових технологій, та основні модулі для розробки додатку.

Метою розробки було створення єдиного стабільного WEB-застосунк який може забезпечити стабільність і довіру, як до національної валюти, так і до віртуальної валюти. Для реалізації даного застосунку було проведено наступні етапи розробки:

- аналіз вимог до програмного забезпечення, проаналізовано та виділено основний набір сервісів. Аналіз популярних рішень, виділені ключові моменти технічних рішень, проведено аналіз архітектурних рішень та основних модулів для розробки (pop-up, mock-api, table). Проаналізовано та виділено основні функції та користувацькі вимоги. Та розписано функціональні вимоги до кожного користувацького кейсу. Було описано нефункціональних вимог. Після проведеного аналізу було поставно комплекс завдань до застосунку інтерфейсу та архітектури.

- конструювання програмного забезпечення. Описані технології, завдяки яким була розроблена система. Було описано діаграми станів для проведення платежів та купівлі валюти. Було описана архітектура додатку. Структура файлової системи та розміщення компонентів, приведені діаграми, документацію з описом ключових модулів додатку, структуру бази даних та детальний опис моделей. Зроблено аналіз безпеки.

- аналіз якості ПЗ. Було описано основні види та механізми тестування. Описано процедуру проведення тестів. Було протестована коректність роботи при різних даних. Впровадження та супровід, описано процедуру розгортання для застосунку та налаштування і запуску додатку на сервері.

- описано Керівництво програміста, а саме описано процедури інсталяції додатку, розгортання, запуск тестів, серверну компіляцію, та норми правлення та фіксування змін.

- описано Керівництво користувача в якому було зображено усі сторінку додатку з описами процедур використання.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440.02.81 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 64 |

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Angular [Електронний ресурс] Режим доступу: <https://angular.io/>
- 2) Adam Wathan and Steve Schoger “Refactoring UI” [Книга, Електронний ресурс] Режим доступу: <https://refactoringui.com/book/>
- 3) Json-server [Електронний ресурс] Режим доступу: <https://www.npmjs.com/package/json-server>
- 4) Pop-up, Alerts [Електронний ресурс] Режим доступу: <https://www.npmjs.com/package/ngx-alerts>
- 5) Table [Електронний ресурс] Режим доступу: <https://www.npmjs.com/package/ngx-magic-table>
- 6) Приватбанк [Електронний ресурс] Режим доступу: <https://www.privat24.ua/>
- 7) Savvy [Електронний ресурс] Режим доступу: <https://www.savvy.io/>
- 8) UXDA [Електронний ресурс] Режим доступу: <https://www.uxdesignagency.com/>
- 9) Канер С. Тестування програмного забезпечення. Фундаментальні концепції менеджменту бізнес-додатків [текст]: /С. Канер; - ДіаСофт, 2001, 544 с, ISBN 966-7393-87-9.
- 10) Калбертсон Р. Швидке тестування [текст]/Р. Калбертсон, К. Браун, Г. Кобб; - Вільямс, 2002, 384 с. ISBN 5-8459-0336-X.
- 11) Autofill [Електронний ресурс] Режим доступу: <https://www.w3.org/TR/html52/sec-forms.html#autofill>
- 12) Бейзер Б. Тестування чорної скриньки [текст]: - Б. Бейзер; - Пітер, 2004, 320 с. ISBN 5-94723-698-2.
- 13) Карла И. Вигерса и Джоя Битти [текст]: Разработка требований к программному обеспечению

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Опис програми

КПІ.ІП-4309.045440-04-13

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 року

Тексти програмного коду

WEB-застосування для управління коштами

(Найменування програми (документа))

CD-R

(Вид носія даних)

132 арк, 283 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ - 2019

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-04-13 | Арк. |
| | | | | | | 2 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |


```

{
  "name": "ch-fe",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "start:bi": "ng serve bank-individual",
    "start:host": "ng serve bank-individual --host=0.0.0.0",
    "start:material": "ng serve material",
    "build": "ng build",
    "build:bi": "ng build bank-individual --prod",
    "build:material": "ng build material",
    "test": "ng test",
    "test:bi": "ng test bank-individual",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~7.2.0",
    "@angular/common": "~7.2.0",
    "@angular/compiler": "~7.2.0",
    "@angular/core": "~7.2.0",
    "@angular/forms": "~7.2.0",
    "@angular/http": "~7.2.0",
    "@angular/platform-browser": "~7.2.0",
    "@angular/platform-browser-dynamic": "~7.2.0",
    "@angular/platform-server": "~7.2.0",
    "@angular/router": "~7.2.0",
    "@mdi/svg": "^3.5.95",
    "@types/chart.js": "^2.7.50",
    "bootstrap": "^4.3.1",
    "chart.js": "^2.8.0",
    "core-js": "^2.5.4",
    "cryptocurrency-icons": "^0.12.0",
    "flag-icon-css": "^3.3.0",
    "ngx-auth": "^5.0.0",
    "rxjs": "~6.3.3",
    "tslib": "^1.9.0",
    "zone.js": "~0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.13.0",
    "@angular-devkit/build-ng-packagr": "~0.13.0",
    "@angular/cli": "~7.3.6",
    "@angular/compiler-cli": "~7.2.0",
    "@angular/language-service": "~7.2.0",
    "@compodoc/compodoc": "^1.1.9",
    "@types/jasmine": "~2.8.8",
    "@types/jasminewd2": "~2.0.3",
    "@types/node": "~8.9.4",
    "animate.css": "^3.7.0",
    "codelyzer": "~4.5.0",
    "jasmine-core": "~2.99.1",
    "jasmine-spec-reporter": "~4.2.1",
    "karma": "~4.0.0",
    "karma-chrome-launcher": "~2.2.0",
    "karma-coverage-istanbul-reporter": "~2.0.1",
    "karma-jasmine": "~1.1.2",
    "karma-jasmine-html-reporter": "^0.2.2",
    "ng-packagr": "^4.2.0",
    "ngx-toastr": "^10.0.2",
    "protractor": "~5.4.0",
    "ts-node": "~7.0.0",
    "tsickle": ">=0.34.0",
    "tslib": "^1.9.0",
    "tslint": "~5.11.0",
    "typescript": "~3.2.2"
  },
  "$schema":
    "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "ch-fe": {
      "root": "",
      "sourceRoot": "src",
      "projectType": "application",
      "prefix": "ch",
      "schematics": {
        "@schematics/angular:component": {
          "style": "scss"
        }
      },
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist/ch-fe",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "src/tsconfig.app.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.scss",
              "node_modules/animate.css/animate.min.css"
            ]
          }
        }
      }
    }
  }
}

```

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-04-13 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

```

    ],
    "scripts": [],
    "es5BrowserSupport": true
  },
  "configurations": {
    "production": {
      "fileReplacements": [
        {
          "replace": "src/environments/environment.ts",
          "with":
"src/environments/environment.prod.ts"
        }
      ],
      "optimization": true,
      "outputHashing": "all",
      "sourceMap": false,
      "extractCss": true,
      "namedChunks": false,
      "aot": true,
      "extractLicenses": true,
      "vendorChunk": false,
      "buildOptimizer": true,
      "budgets": [
        {
          "type": "initial",
          "maximumWarning": "2mb",
          "maximumError": "5mb"
        }
      ]
    }
  },
  "serve": {
    "builder": "
"@angular-devkit/build-angular:dev-
server",

    "options": {
      "browserTarget": "ch-fe:build"
    },
    "configurations": {
      "production": {
        "browserTarget": "ch-fe:build:production"
      }
    }
  },
  "extract-i18n": {
    "builder": "
"@angular-devkit/build-angular:extract-
i18n",

    "options": {
      "browserTarget": "ch-fe:build"
    }
  },
  "test": {
    "builder": "@angular-devkit/build-angular:karma",
    "options": {
      "main": "src/test.ts",
      "polyfills": "src/polyfills.ts",
      "tsConfig": "src/tsconfig.spec.json",
      "karmaConfig": "src/karma.conf.js",
      "styles": [
        "src/styles.scss"
      ],
      "scripts": [],
      "assets": [
        "src/favicon.ico",
        "src/assets"
      ]
    }
  },
  "lint": {
    "builder": "@angular-devkit/build-angular:tslint",
    "options": {
      "tsConfig": [
        "src/tsconfig.app.json",
        "src/tsconfig.spec.json"
      ],
      "exclude": [
        "**/node_modules/**"
      ]
    }
  },
  "ch-fe-e2e": {
    "root": "e2e/",
    "projectType": "application",
    "prefix": "",
    "architect": {
      "e2e": {
        "builder": "
"@angular-devkit/build-
angular:protractor",

        "options": {
          "protractorConfig": "e2e/protractor.conf.js",
          "devServerTarget": "ch-fe:serve"
        },
        "configurations": {
          "production": {

```

```

    "devServerTarget": "ch-fe:serve:production"
  }
},
"lint": {
  "builder": "@angular-devkit/build-angular:tslint",
  "options": {
    "tsConfig": "e2e/tsconfig.e2e.json",
    "exclude": [
      "**/node_modules/**"
    ]
  }
},
"bank-individual": {
  "root": "projects/bank-individual/",
  "sourceRoot": "projects/bank-individual/src",
  "projectType": "application",
  "prefix": "bi",
  "schematics": {
    "@schematics/angular:component": {
      "style": "scss"
    }
  },
  "architect": {
    "build": {
      "builder": "@angular-devkit/build-
angular:browser",
      "options": {
        "outputPath": "dist/bank-individual",
        "index": "projects/bank-individual/src/index.html",
        "main": "projects/bank-individual/src/main.ts",
        "polyfills": "projects/bank-
individual/src/polyfills.ts",
        "tsConfig": "projects/bank-
individual/tsconfig.app.json",
        "assets": [
          "projects/bank-individual/src/favicon.ico",
          "projects/bank-individual/src/assets",
          {
            "glob": "**/*",
            "input": "./node_modules/@mdi/svg/svg",
            "output": "./assets/icons"
          },
          {
            "glob": "**/*",
            "input": "./node_modules/flag-icon-
css/flags/1x1",
            "output": "./assets/icons/flags"
          },
          {
            "glob": "**/*",
            "input": "./node_modules/cryptocurrency-
icons/128/icon",
            "output": "./assets/icons/crypto"
          }
        ],
        "styles": [
          "projects/bank-individual/src/styles/styles.scss",
          "node_modules/animate.css/animate.min.css"
        ],
        "scripts": [],
        "es5BrowserSupport": true
      },
      "configurations": {
        "production": {
          "fileReplacements": [
            {
              "replace": "projects/bank-
individual/src/environments/environment.ts",
              "with": "projects/bank-
individual/src/environments/environment.prod.ts"
            }
          ],
          "optimization": true,
          "outputHashing": "all",
          "sourceMap": false,
          "extractCss": true,
          "namedChunks": false,
          "aot": true,
          "extractLicenses": true,
          "vendorChunk": false,
          "buildOptimizer": true,
          "budgets": [
            {
              "type": "initial",
              "maximumWarning": "2mb",
              "maximumError": "5mb"
            }
          ]
        }
      },
      "serve": {

```

```
"builder": "@angular-devkit/build-angular:dev-
server",
"options": {
  "browserTarget": "bank-individual:build"
},
"configurations": {
  "production": {
    "browserTarget": "bank-
individual:build:production"
  }
},
"extract-i18n": {
  "builder": "@angular-devkit/build-angular:extract-
i18n",
  "options": {
    "browserTarget": "bank-individual:build"
  }
},
"test": {
  "builder": "@angular-devkit/build-angular:karma",
  "options": {
    "main": "projects/bank-individual/src/test.ts",
    "polyfills": "projects/bank-
individual/src/polyfills.ts",
    "tsConfig": "projects/bank-
individual/tsconfig.spec.json",
    "karmaConfig": "projects/bank-
individual/karma.conf.js",
    "styles": [
      "projects/bank-individual/src/styles/styles.scss"
    ],
    "scripts": [],
    "assets": [
      "projects/bank-individual/src/favicon.ico",
      "projects/bank-individual/src/assets"
    ]
  }
},
"lint": {
  "builder": "@angular-devkit/build-angular:tslint",
  "options": {
    "tsConfig": [
      "projects/bank-individual/tsconfig.app.json",
      "projects/bank-individual/tsconfig.spec.json"
    ],
    "exclude": [
      "**/node_modules/**"
    ]
  }
}
}
},
"server": {
  "builder": "@angular-devkit/build-angular:server",
  "options": {
    "outputPath": "dist/bank-individual-server",
    "main": "projects/bank-
individual/src/main.server.ts",
    "tsConfig": "projects/bank-
individual/tsconfig.server.json"
  },
  "configurations": {
    "production": {
      "fileReplacements": [
        {
          "replace": "projects/bank-
individual/src/environments/environment.ts",
          "with": "projects/bank-
individual/src/environments/environment.prod.ts"
        }
      ],
      "sourceMap": false,
      "optimization": {
        "scripts": false,
        "styles": true
      }
    }
  },
  "app-shell": {
    "builder": "@angular-devkit/build-angular:app-
shell",
    "options": {
      "browserTarget": "bank-individual:build",
      "serverTarget": "bank-individual:server",
      "route": "shell"
    },
    "configurations": {
      "production": {
        "browserTarget": "bank-
individual:build:production",
        "serverTarget": "bank-
individual:server:production"
      }
    }
  }
}
```

```

    },
    "bank-individual-e2e": {
      "root": "projects/bank-individual-e2e/",
      "projectType": "application",
      "prefix": "",
      "architect": {
        "e2e": {
          "builder": "@angular-devkit/build-
angular:protractor",
          "options": {
            "protractorConfig": "projects/bank-individual-
e2e/protractor.conf.js",
            "devServerTarget": "bank-individual:serve"
          },
          "configurations": {
            "production": {
              "devServerTarget": "bank-
individual:serve:production"
            }
          }
        },
        "lint": {
          "builder": "@angular-devkit/build-angular:tslint",
          "options": {
            "tsConfig": "projects/bank-individual-
e2e/tsconfig.e2e.json",
            "exclude": [
              "**/node_modules/**"
            ]
          }
        }
      },
      "pop-up": {
        "root": "libs/pop-up",
        "sourceRoot": "libs/pop-up/src",
        "projectType": "library",
        "prefix": "lib",
        "architect": {
          "build": {
            "builder": "@angular-devkit/build-ng-
packagr:build",
            "options": {
              "tsConfig": "libs/mock-api/tsconfig.lib.json",
              "project": "libs/mock-api/ng-package.json"
            }
          },
          "test": {
            "builder": "@angular-devkit/build-angular:karma",
            "options": {
              "main": "libs/mock-api/src/test.ts",
              "tsConfig": "libs/mock-api/tsconfig.spec.json",
              "karmaConfig": "libs/mock-api/karma.conf.js"
            }
          },
          "lint": {
            "builder": "@angular-devkit/build-angular:tslint",
            "options": {
              "tsConfig": [
                "libs/mock-api/tsconfig.lib.json",

```

```

      "libs/mock-api/tsconfig.spec.json"
    ],
    "exclude": [
      "**/node_modules/**"
    ]
  }
},
"bank-cms": {
  "root": "projects/bank-cms/",
  "sourceRoot": "projects/bank-cms/src",
  "projectType": "application",
  "prefix": "bi",
  "schematics": {
    "@schematics/angular:component": {
      "style": "scss"
    }
  },
  "architect": {
    "build": {
      "builder": "@angular-devkit/build-
angular:browser",
      "options": {
        "outputPath": "dist/bank-cms",
        "index": "projects/bank-cms/src/index.html",
        "main": "projects/bank-cms/src/main.ts",
        "polyfills": "projects/bank-cms/src/polyfills.ts",
        "tsConfig": "projects/bank-cms/tsconfig.app.json",
        "assets": [
          "projects/bank-cms/src/favicon.ico",
          "projects/bank-cms/src/assets"
        ],
        "styles": [
          "projects/bank-cms/src/styles.scss"
        ],
        "scripts": [],
        "es5BrowserSupport": true
      },
      "configurations": {
        "production": {
          "fileReplacements": [
            {
              "replace": "projects/bank-
cms/src/environments/environment.ts",
              "with": "projects/bank-
cms/src/environments/environment.prod.ts"
            }
          ],
          "optimization": true,
          "outputHashing": "all",
          "sourceMap": false,
          "extractCss": true,
          "namedChunks": false,
          "aot": true,
          "extractLicenses": true,
          "vendorChunk": false,
          "buildOptimizer": true,
          "budgets": [
            {
              "type": "initial",
              "maximumWarning": "2mb",
              "maximumError": "5mb"
            }
          ]
        }
      },
      "serve": {
        "builder": "@angular-devkit/build-angular:dev-
server",
        "options": {
          "browserTarget": "bank-cms:build"
        },
        "configurations": {
          "production": {
            "browserTarget": "bank-cms:build:production"
          }
        }
      },
      "extract-i18n": {
        "builder": "@angular-devkit/build-angular:extract-
i18n",
        "options": {
          "browserTarget": "bank-cms:build"
        }
      },
      "test": {
        "builder": "@angular-devkit/build-angular:karma",
        "options": {
          "main": "projects/bank-cms/src/test.ts",
          "polyfills": "projects/bank-cms/src/polyfills.ts",
          "tsConfig": "projects/bank-
cms/tsconfig.spec.json",
          "karmaConfig": "projects/bank-
cms/karma.conf.js",

```

```

"styles": [
  "projects/bank-cms/src/styles.scss"
],
"scripts": [],
"assets": [
  "projects/bank-cms/src/favicon.ico",
  "projects/bank-cms/src/assets"
]
},
"lint": {
  "builder": "@angular-devkit/build-angular:tslint",
  "options": {
    "tsConfig": [
      "projects/bank-cms/tsconfig.app.json",
      "projects/bank-cms/tsconfig.spec.json"
    ],
    "exclude": [
      "**/node_modules/**"
    ]
  }
},
"bank-cms-e2e": {
  "root": "projects/bank-cms-e2e/",
  "projectType": "application",
  "prefix": "",
  "architect": {
    "e2e": {
      "builder": "@angular-devkit/build-
angular:protractor",
      "options": {
        "protractorConfig": "projects/bank-cms-
e2e/protractor.conf.js",
        "devServerTarget": "bank-cms:serve"
      },
      "configurations": {
        "production": {
          "devServerTarget": "bank-cms:serve:production"
        }
      }
    },
    "lint": {
      "builder": "@angular-devkit/build-angular:tslint",
      "options": {
        "tsConfig": "projects/bank-cms-
e2e/tsconfig.e2e.json",

```

```

"exclude": [
  "**/node_modules/**"
]
},
"defaultProject": "ch-fe"
},
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "sourceMap": true,
    "declaration": false,
    "module": "es2015",
    "moduleResolution": "node",
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "importHelpers": true,
    "target": "es5",
    "typeRoots": [
      "node_modules/@types"
    ],
    "lib": [
      "es2018",
      "es6",
      "dom"
    ],
    "paths": {
      "pop-up": [
        "dist/pop-up"
      ],
      "pop-up/*": [
        "dist/pop-up/*"
      ],
      "mock-api": [
        "dist/mock-api"
      ],
      "mock-api/*": [
        "dist/mock-api/*"
      ]
    }
  }
}

```

```
export const MONTHS_NUMBERS: string[] = [
  '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12',
];

export const CURRENT_YEAR = new
Date().getFullYear();

export const MAX_YEAR = CURRENT_YEAR + 100;

export const YEARS_NUMBERS: number[] = () => {
  const length = MAX_YEAR - CURRENT_YEAR;
  const years: number[] = new Array(length);
  for (let i = 0; i < length; i++) {
    years[i] = CURRENT_YEAR + i;
  }
  return years;
})();

export const GENDERS_LIST = [
  'MALE',
  'FEMALE'
];

export const GENDERS_OBJECT = {
  M: 'MALE',
  F: 'FEMALE'
};

export const GENDERS_DICTIONARY = [
  { value: 'MALE', key: 'M' },
  { value: 'FEMALE', key: 'F' },
];

import { Observable, Subject, Subscription } from 'rxjs';
import { of } from 'rxjs/internal/observable/of';

export function cacheObsSubject<T>(args?: any):
Subject<T> {
  let data: T = null;
  const source: Observable<T> = (args ? new
Observable<T>(args) : new Observable<T>());
  //We create a Subject, and pass it as an observer
  //to the source so that it will receive all the values pushed
  out by this source
  const subject: Subject<T> = new Subject<T>();
  const subSource = source.subscribe(subject);
  return subject;
}
```

```

}
export function cacheObservable<T>(): () =>
Observable<T> {
  let data: T = null;
  let timeLeft = 0;
  const subject: Subject<T> = new Subject<T>();
  const subjectObservable: Observable<T> =
subject.asObservable();
  const sourceObservable = this;
  let subscription: Subscription =
sourceObservable.subscribe((res) => {
    data = res;
    subject.next(data);
  });

  return ((refresh?: boolean, forceDate?: any) => {
    // force update
    if (forceDate) {
      data = forceDate;
      return of(data);
    }
    // update
    if (refresh) {
      timeLeft = 0;
      data = null;
      subscription.unsubscribe();
      subscription = sourceObservable.subscribe((res) => {
        data = res;
        subject.next(data);
      });
    }
    // return loaded date
    if (data) {
      return of(data);
    }
    // then data are not received or loading
    return subjectObservable;
  });
}

export function cachePeriodObservable<T>(count:
number): (refresh: boolean | undefined) => Observable<T> {
  let data: T = null;
  let timeLeft = 0;
  const subject: Subject<T> = new Subject<T>();
  const sourceObservable = this;
  let subscription: Subscription = this.subscribe((res) => {

```



```

data = res;
subject.next(data);
});

return ((refresh: boolean, forceDate?: any) => {
  // force update
  if (forceDate) {
    data = forceDate;
    return of(data);
  }
  // update
  if (refresh || ++timeLeft > count) {
    timeLeft = 0;
    data = null;
    subscription.unsubscribe();
    subscription = sourceObservable.subscribe((res) => {
      data = res;
      subject.next(data);
    });
  }
  // return loaded date
  if (data) {
    return of(data);
  }
  // then data are not received or loading
  return subject;
});
}

export function calculateOffset(obj: any): { x: number, y:
number } {
  let x = 0;
  let y = 0;

  if (obj.offsetParent) {
    do {
      x += obj.offsetLeft;
      y += obj.offsetTop;
      obj = obj.offsetParent;
    } while (obj);
  }

  return {x, y};
}

export function changePasswordTypeText(input: any):
string {

return input.type = input.type === 'password' ? 'text' :
'password';
}

import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';

function _containsItem<T>(array: T[], item: T): boolean {
  for (const arrayItem of array) {
    if (String(arrayItem) === String(item)) {
      return true;
    }
  }
  return false;
}

function _containsItems<T>(array: T[], items: T[]):
boolean {
  const cans: boolean[] = Array(items.length).fill(false, 0,
items.length);
  for (const arrayItem of array) {
    for (let i = 0; i < items.length; i++) {
      if (String(arrayItem) === String(items[i])) {
        cans[i] = true;
      }
    }
  }
  return cans.reduce((r, n) => r && n, true);
}

function _containsAnyItems<T>(array: T[], items: T[]):
boolean {
  for (const arrayItem of array) {
    for (const item of items) {
      if (String(arrayItem) === String(item)) {
        return true;
      }
    }
  }

  return false;
}

function _observableContainsItem<T>(observableArray:
Observable<T[]>, items: T): Observable<boolean> {
  return observableArray.pipe(map((array) =>
_containsItem(array, items)));
}

```

```
function _observableContainsItems<T>(observableArray:
Observable<T[]>, items: T[]): Observable<boolean> {
    return observableArray.pipe(map((array) =>
    _containsItems(array, items)));
}

function
_observableContainsAnyItems<T>(observableArray:
Observable<T[]>, items: T[]): Observable<boolean> {
    return observableArray.pipe(map((array) =>
    _containsAnyItems(array, items)));
}

export function containsItems<T>(array:
Observable<T[]> | T[], item: T | T[])
: Observable<boolean> | boolean | undefined {
    if (!array || !item) {
        return undefined;
    }

    if (array instanceof Observable) {
        if (Array.isArray(item)) {
            return _observableContainsItems(array, item);
        } else {
            return _observableContainsItem(array, item);
        }
    }

    if (Array.isArray(array)) {
        if (Array.isArray(item)) {
            return _containsItems(array, item);
        } else {
            return _containsItem(array, item);
        }
    }

    return undefined;
}

export function containsAnyItems<T>(array:
Observable<T[]> | T[], item: T | T[])
: Observable<boolean> | boolean | undefined {
    if (!array || !item) {
        return undefined;
    }
}
```

```
if (array instanceof Observable) {
    if (Array.isArray(item)) {
        return _observableContainsAnyItems(array, item);
    } else {
        return _observableContainsItem(array, item);
    }
}

if (Array.isArray(array)) {
    if (Array.isArray(item)) {
        return _containsAnyItems(array, item);
    } else {
        return _containsItem(array, item);
    }
}

return undefined;
}

/**
 * @param date template: DD.MM.YYYY
hour:minute:second
 */
export function dateFromAPIDateString(date: string |
Date): Date {
    if (!date) {
        return null;
    }

    const t: string[] = date.toString().split(/[. :]/);

    let DD = +t[0];
    let MM = +t[1] - 1;
    let YYYY = +t[2];

    let HH = +t[3];
    let mm = +t[4];
    let ss = +t[5];

    DD = DD ? DD : 0;
    MM = MM ? MM : 0;
    YYYY = YYYY ? YYYY : 0;

    HH = HH ? HH : 0;
    mm = mm ? mm : 0;
    ss = ss ? ss : 0;

    return new Date(YYYY, MM, DD, HH, mm, ss);
}
```

```
/**
 * @returns string DD-MM-YYYY hour:minute:second
 */
export function dateToAPIString(date: string | Date, time
= true): string {
  if (date instanceof Date) {
    const dd = date.getDate();
    const month = +date.getMonth() + 1;
    const YYYY = date.getFullYear();
    const hh = date.getHours();
    const mm = date.getMinutes();
    const ss = date.getSeconds();

    const DD = ((dd) < 10) ? '0' + dd : dd;
    const MM = ((month) < 10) ? '0' + month : month;

    const HH = ((hh) < 10) ? '0' + hh : hh;
    const min = ((mm) < 10) ? '0' + mm : mm;
    const SS = ((ss) < 10) ? '0' + ss : ss;

    let text = `${YYYY}-${MM}-${DD}`;
    if (time) {
      text += ` ${HH}:${min}:${SS}`;
    }
    return text;
  }
  return null;
}

export const TODAY = new Date();

export enum DatePeriods {
  TODAY,
  YESTERDAY,
  LAST_WEEK,
  CURRENT_MONTH,
  CURRENT_MONTH_PREVIOUS_DAY,
  PREVIOUS_MONTH,
  FROM_PREVIOUS_MONTH,
  CURRENT_QUARTER,
  PREVIOUS_QUARTER,
  CURRENT_YEAR,
}

export function getToday(): { from: Date; to: Date; } {
  const start = new Date();
  const end = new Date();

  start.setHours(0, 0, 0, 0);
  end.setHours(23, 59, 59, 999);

  return {
    from: start,
    to: end,
  };
}

export function getYesterday(): { from: Date; to: Date; } {
  const start = new Date();
  const yesterdayEnd = new Date();
  start.setDate(start.getDate() - 1);
  yesterdayEnd.setDate(yesterdayEnd.getDate() - 1);
  start.setHours(0, 0, 0, 0);
  yesterdayEnd.setHours(23, 59, 59, 999);
  return {
    from: start,
    to: yesterdayEnd,
  };
}

export function getCurrentWeek(): { from: Date; to: Date; } {
  const start = new Date();
  const today = new Date();

  start.setDate(start.getDate() - start.getDay() -
(start.getDay() === 0 ? 6 : -1));

  start.setHours(0, 0, 0, 0);
  today.setHours(23, 59, 59, 999);
  return {
    from: start,
    to: today,
  };
}

export function getLastWeek(): { from: Date; to: Date; } {
  const start = new Date();
  const today = new Date();

  start.setDate(start.getDate() - 7);

  start.setHours(0, 0, 0, 0);
  today.setHours(23, 59, 59, 999);
  return {
    from: start,
    to: today,
  };
}
```

```
};
}

export function getCurrentMonth(): { from: Date; to:
Date; } {
    const start = new Date();
    const today = new Date();
    start.setDate(1);
    start.setHours(0, 0, 0, 0);
    today.setHours(23, 59, 59, 999);
    return {
        from: start,
        to: today,
    };
}

export function getCurrentMonthPreviousDay(): { from:
Date; to: Date; } {
    const start = new Date();
    const yesterday = new Date();
    yesterday.setDate(yesterday.getDate() - 1);
    start.setDate(1);
    start.setHours(0, 0, 0, 0);
    return {
        from: start,
        to: yesterday,
    };
}

export function getPreviousMonth(): { from: Date; to:
Date; } {
    const start = new Date();
    const mouthEnd = new Date();
    start.setMonth(start.getMonth() - 1);
    start.setDate(1);
    mouthEnd.setDate(0);
    start.setHours(0, 0, 0, 0);
    mouthEnd.setHours(23, 59, 59, 999);
    return {
        from: start,
        to: mouthEnd,
    };
}

export function getFromPreviousMonth(): { from: Date;
to: Date; } {
    const start = new Date();
    const mouthEnd = new Date();

    start.setMonth(start.getMonth() - 1);
    start.setDate(1);
    mouthEnd.setHours(23, 59, 59, 999);
    return {
        from: start,
        to: mouthEnd,
    };
}

export function getCurrentQuarter(): { from: Date; to:
Date; } {
    const now = new Date();
    const quarter = Math.floor((new Date().getMonth() + 3) /
3);
    const firstDate = new Date(now.getFullYear(), quarter *
3 - 3, 1);
    const endDate = new Date();
    firstDate.setHours(0, 0, 0, 0);
    endDate.setHours(23, 59, 59, 999);
    return {
        from: firstDate,
        to: endDate,
    };
}

export function getPreviousQuarter(): { from: Date; to:
Date; } {
    const now = new Date();
    const quarter = Math.floor((new Date().getMonth() + 3) /
3) - 1;
    const firstDate = new Date(now.getFullYear(), quarter *
3 - 3, 1);
    const endDate = new Date(firstDate.getFullYear(),
firstDate.getMonth() + 3, 0);
    firstDate.setHours(0, 0, 0, 0);
    endDate.setHours(23, 59, 59, 999);
    return {
        from: firstDate,
        to: endDate,
    };
}

export function getCurrentYear(): { from: Date; to: Date;
} {
    const date = new Date();
```

```
const firstDay = new Date(date.getFullYear(), 0, 1);
const today = new Date();
firstDay.setHours(0, 0, 0, 0);
today.setHours(23, 59, 59, 999);

return {
  from: firstDay,
  to: today,
};
}
```

```
export function getDateRange(datePeriod: DatePeriods):
[Date, Date] {
  switch (datePeriod) {
    case DatePeriods.TODAY: {
      const range = getToday();
      return [range.from, range.to];
    }
    case DatePeriods.YESTERDAY: {
      const range = getYesterday();
      return [range.from, range.to];
    }
    case DatePeriods.CURRENT_MONTH: {
      const range = getCurrentMonth();
      return [range.from, range.to];
    }
    case
DatePeriods.CURRENT_MONTH_PREVIOUS_DAY: {
      const range = getCurrentMonthPreviousDay();
      return [range.from, range.to];
    }
    case DatePeriods.PREVIOUS_MONTH: {
      const range = getPreviousMonth();
      return [range.from, range.to];
    }
    case DatePeriods.FROM_PREVIOUS_MONTH: {
      const range = getFromPreviousMonth();
      return [range.from, range.to];
    }
    case DatePeriods.CURRENT_QUARTER: {
      const range = getCurrentQuarter();
      return [range.from, range.to];
    }
    case DatePeriods.PREVIOUS_QUARTER: {
      const range = getPreviousQuarter();
      return [range.from, range.to];
    }
  }
}
```

```
case DatePeriods.LAST_WEEK: {
  const range = getLastWeek();
  return [range.from, range.to];
}
case DatePeriods.CURRENT_YEAR: {
  const range = getCurrentYear();
  return [range.from, range.to];
}
}
```

```
export function filterFnDate(data: Date, filter: Date[]):
boolean {
  if (data && filter && filter.length >= 2) {
    return data >= filter[0] && data <= filter[1];
  }
  return true;
}

export function filterDates(key: string,
data: { [key: string]: Date }[],
filtes: Date[],
filterFn: (data: Date, filter: Date[]) =>
boolean = filterFnDate): any[] {
  return data.filter((i) => filterFn(i[key], filtes));
}
```

```
export enum CardType {
  VISA = 'VISA',
  MASTERCARD = 'MASTERCARD',
  ELECTRON = 'ELECTRON',
  MAESTRO = 'MAESTRO',
  DANKORT = 'DANKORT',
  INTERPAYMENT = 'INTERPAYMENT',
  UNIONPAY = 'UNIONPAY',
  AMEX = 'AMEX',
  DINERS = 'DINERS',
  DISCOVER = 'DISCOVER',
  JCB = 'JCB',
}
```

```
export function detectCardType(numbr: string): CardType
{
  const re = {
    ELECTRON:
/^(4026|417500|4405|4508|4844|4913|4917)\d+$/,
```

```

MAESTRO:
/^(5018|5020|5038|5612|5893|6304|6759|6761|6762|6763|0604|6390)\
d+$/,

DANKORT: /^(5019)\d+$/,
INTERPAYMENT: /^(636)\d+$/,
UNIONPAY: /^(62|88)\d+$/,
VISA: /^[0-9]{12}(\?:[0-9]{3})?$/,
MASTERCARD: /^[5|1-5][0-9]{14}$/,
AMEX: /^[47][0-9]{13}$/,
DINERS: /^(?:0[0-5]||[68][0-9])[0-9]{11}$/,
DISCOVER: /^6(?:011|5[0-9]{2})[0-9]{12}$/,
JCB: /^(?:2131|1800|35\d{3})\d{11}$/

};

for (const key in re) {
  if (re[key].test(numbr)) {
    return CardType[key];
  }
}

// +380 44 xxx-xx-xx (international call to Kiev)
// +380 45 94x-xx-xx (international call to Brovary)
// +380 48 xxx-xx-xx (international call to Odessa)
// +380 48 2xx-xx-xx (international call to 6-digit
numbers in Odessa)

// +380 32 xxx-xx-xx (international call to Lviv)
// +380 32 2xx-xx-xx (international call to 6-digit
numbers in Lviv)

export enum Operators {
  KYIVSTAR,
  LIFECCELL,
  VODAFONE,
  THREE_MOB,
  PEOPLENET,
  INTERTELECOM,
}

export class OperatorMap {
  50: Operators.VODAFONE;
  95: Operators.VODAFONE;
  66: Operators.VODAFONE;
  99: Operators.VODAFONE;

  63: Operators.LIFECCELL;
  73: Operators.LIFECCELL;
  93: Operators.LIFECCELL;

  68: Operators.KYIVSTAR;
  67: Operators.KYIVSTAR;
  96: Operators.KYIVSTAR;
  97: Operators.KYIVSTAR;
  98: Operators.KYIVSTAR;

  91: Operators.THREE_MOB;
  92: Operators.PEOPLENET;
  94: Operators.INTERTELECOM;
}

export function detectOperator(mNumber: string):
Operators {
  if (mNumber.startsWith('+')) {
    mNumber = mNumber.substr(1, mNumber.length - 1);
  }
  if (mNumber.startsWith('380')) {
    mNumber = mNumber.substr(3, mNumber.length - 3);
  }

  return OperatorMap[mNumber.substr(0, 2)];
}

export function generateUniqueId(length: number): string
{
  let idHash = "";
  const possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

  for (let i = 0; i < length; i++) {
    idHash += possible.charAt(Math.floor(Math.random()
* possible.length));
  }
  return idHash;
}

export function isIE(): boolean {
  return (navigator.appName === 'Microsoft Internet
Explorer'
|| !(navigator.userAgent.match(/Trident/)
|| navigator.userAgent.match(/rv:11/)));
}
```

```

export function shuffle(array: any[]): void {
  for (let i = 0; i < array.length; i++) {
    const tmp = array[i];
    const r = Math.floor(Math.random() * (array.length - i)
+ i);

    array[i] = array[r];
    array[r] = tmp;
  }
}

const toCamel = (s) => {
  return s.replace(/([-_][a-z])/ig, ($1) => {
    return $1.toUpperCase()
      .replace('-', '')
      .replace('_', '');
  });
};

const isArray = (a) => {
  return Array.isArray(a);
};

const isObject = (o) => {
  return o === Object(o) && !isArray(o) && typeof o !==
'function';
};

export function snakeToCamel(o) {
  if (isObject(o)) {
    const n = {};

    Object.keys(o)
      .forEach(k => {
        n[toCamel(k)] = snakeToCamel(o[k]);
      });

    return n;
  } else if (isArray(o)) {
    return o.map((i) => {
      return snakeToCamel(i);
    });
  }

  return o;
}

const firstLetters = { };

```

```

for (const p in unicodeCategories) {
  if (firstLetters[p[0]]) {
    firstLetters[p[0]] = unicodeCategories[p].substring(0,
unicodeCategories[p].length - 1) + firstLetters[p[0]].substring(1);
  } else {
    firstLetters[p[0]] = unicodeCategories[p];
  }
}

for (const p in firstLetters) {
  unicodeCategories[p] = firstLetters[p];
}

/* \p{L&} aka \p{Cased_Letter} - <http://www.regular-
expressions.info/unicode.html#category>

By
<http://stackoverflow.com/users/394921/lwngaman>

From this StackOverflow question:
<http://stackoverflow.com/q/280712>

*/

unicodeCategories['L&'] =
unicodeCategories['Lu'].substring(0, unicodeCategories['Lu'].length -
1) +
unicodeCategories['Ll'].substring(1,
unicodeCategories['Ll'].length - 1) +
unicodeCategories['Lt'].substring(1, unicodeCategories['Lt'].length);

/* Gets a regex written in a dialect that supports unicode
categories and
  translates it to a dialect supported by JavaScript. */
return function (regexString) {
  let modifiers = "";
  if (regexString instanceof RegExp) {
    modifiers = (regexString.global ? 'g' : "") +
      (regexString.ignoreCase ? 'i' : "") +
      (regexString.multiline ? 'm' : "");
    regexString = regexString.source;
  }
  regexString = regexString.replace(/\p{(.?)}/g,
function (match, group) {
    return unicodeCategories[group] || match;
  });
  return new RegExp(regexString, modifiers);
})();
// tslint:enable

import {FormControl, FormGroup} from
'@angular/forms';

```

```

export function validateFormGroupControls(formGroup:
FormGroup): void {
  for (const prop in formGroup.controls) {
    if (formGroup.controls.hasOwnProperty(prop)) {
      const formControl = formGroup.controls[prop];
      if (formControl instanceof FormControl) {
        formControl.markAsDirty({onlySelf: true});
        formControl.markAsTouched({onlySelf: true});
      } else if (formControl instanceof FormGroup) {
        validateFormGroupControls(formControl);
      }
    }
  }
}

```

```

import { Directive, ElementRef, HostListener, Input,
OnDestroy, OnInit, TemplateRef } from '@angular/core';
import { calculateOffset } from
'./../functions/calculateOffset';
import { HeaderService } from '../header.service';
import { HeaderComponentLocation } from '../iheader';

```

```

@Directive({
  selector: '[ksStickToHeader]',
})
export class StickToHeaderDirective implements OnInit,
OnDestroy {

```

```

  @Input('ksStickToHeader') public location:
HeaderContainerLocation;
  @Input() public template: TemplateRef<any>;

```

```

  constructor(private readonly _headerService:
HeaderService,
    private readonly _elementRef: ElementRef) {
    this.location = HeaderComponentLocation.START;
    document.body.children[0].addEventListener('scroll',
this.onScroll.bind(this));
  }

```

```

  public ngOnInit(): void {
    if (!this.location) {
      this.location = HeaderComponentLocation.START;
    }
  }

```

```

  @HostListener('window:scroll')
  public onScroll(): void {

```

```

    const scrolled = window.pageYOffset ||
document.documentElement.scrollTop
    || document.body.children[0].scrollTop || 0;

    const nHeight =
calculateOffset(this._elementRef.nativeElement).y;

    if (scrolled > nHeight) {
      this._headerService.place(this.template, this.location);
    } else {
      this._headerService.remove(this.template);
    }
  }

```

```

  public ngOnDestroy(): void {
    this._headerService.remove(this.template);

```

```

    document.body.children[0].removeEventListener('scroll',
this.onScroll.bind(this));
  }
}

```

```

<header class="container-fluid" [ngClass]="{ scrolled:
!!scrolled }">

```

```

<ng-content *ngTemplateOutlet="start"></ng-content>

```

```

<div class="d-flex" style="flex-grow: 1"></div>

```

```

<ng-content *ngTemplateOutlet="middle"></ng-
content>

```

```

<div class="d-flex" style="flex-grow: 1"></div>

```

```

<ng-content *ngTemplateOutlet="end"></ng-content>
</header>

```

```

import { async, ComponentFixture, TestBed } from
'@angular/core/testing';

```

```

import { RouterTestingModule } from
'@angular/router/testing';

```

```

import { HeaderComponent } from '../header.component';
import { HeaderService } from '../header.service';

```

```

describe('HeaderComponent', () => {
  let component: HeaderComponent;
  let fixture: ComponentFixture<HeaderComponent>;

```



```
beforeEach(async() => {
  TestBed.configureTestingModule({
    declarations: [HeaderComponent],
    imports: [
      RouterTestingModule.withRoutes([]),
    ],
    providers: [
      HeaderService,
    ],
  })
  .compileComponents();
});

beforeEach(() => {
  fixture =
  TestBed.createComponent(HeaderComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should be created', () => {
  expect(component).toBeTruthy();
});

import { Component, HostListener, OnDestroy,
TemplateRef } from '@angular/core';
import { HeaderService } from './header.service';
import { HeaderComponentLocation, IHeader } from
'./iheader';

@Component({
  selector: 'ks-header',
  styleUrls: ['./header.component.scss'],
  templateUrl: './header.component.html',
})
export class HeaderComponent implements IHeader,
OnDestroy {

  public scrolled: boolean;

  public start: TemplateRef<any>;
  public middle: TemplateRef<any>;
  public end: TemplateRef<any>;

  constructor(private readonly _headerService:
HeaderService) {
    this.scrolled = false;

    this._headerService.injectHeader(this);
    document.body.children[0].addEventListener('scroll',
this.onScroll.bind(this));
  }

  public ngOnDestroy(): void {
    this._headerService.cleanHeader();

    document.body.children[0].removeEventListener('scroll',
this.onScroll.bind(this));
  }

  public passRef(templateRef: TemplateRef<any>, hcl:
HeaderContainerLocation): void {
    switch (hcl) {
      case HeaderComponentLocation.START: {
        this.start = templateRef;
        break;
      }
      case HeaderComponentLocation.MIDDLE: {
        this.middle = templateRef;
        break;
      }
      case HeaderComponentLocation.END: {
        this.end = templateRef;
        break;
      }
    }
  }

  public remove(templateRef: TemplateRef<any>): void {
    if (this.start === templateRef) {
      this.start = null;
    }
    if (this.middle === templateRef) {
      this.middle = null;
    }
    if (this.end === templateRef) {
      this.end = null;
    }
  }

  @HostListener('window:scroll')
  public onScroll(): void {
    this.scrolled = (window.pageYOffset
    || document.body.children[0].scrollTop
    || 0) > 0;
  }
}
```

```

    }
  }

  import { CommonModule } from '@angular/common';
  import { ModuleWithProviders, NgModule } from '@angular/core';

  import { StickToHeaderDirective } from './directives/stick-to-header.directive';
  import { HeaderComponent } from './header.component';
  import { HeaderService } from './header.service';

  @NgModule({
    declarations: [
      HeaderComponent,
      StickToHeaderDirective,
    ],
    exports: [
      HeaderComponent,
      StickToHeaderDirective,
    ],
    imports: [
      CommonModule,
    ],
  })
  export class HeaderModule {
    public static forRoot(): ModuleWithProviders {
      return {
        ngModule: HeaderModule,
        providers: [HeaderService],
      };
    }
  }

  import { inject, TestBed } from '@angular/core/testing';

  import { HeaderService } from './header.service';

  describe('HeaderService', () => {
    beforeEach(() => {
      TestBed.configureTestingModule({
        providers: [HeaderService],
      });
    });

    it('should be created', inject([HeaderService], (service: HeaderService) => {
      expect(service).toBeTruthy();
    }));
  });

  import { Injectable, TemplateRef } from '@angular/core';
  import { HeaderContainerLocation, IHeader } from './iheader';

  @Injectable()
  export class HeaderService {
    private header: IHeader;

    constructor() {
    }

    public injectHeader(header: IHeader): void {
      this.header = header;
    }

    public place(templateRef: TemplateRef<any>, hcl: HeaderContainerLocation): void {
      if (this.header) {
        this.header.passRef(templateRef, hcl);
      }
    }

    public remove(templateRef: TemplateRef<any>): void {
      if (this.header) {
        this.header.remove(templateRef);
      }
    }

    public cleanHeader(): void {
      this.header = null;
    }
  }

  import { TemplateRef } from '@angular/core';

  export enum HeaderContainerLocation {
    START = 1,
    MIDDLE,
    END,
  }

  export interface IHeader {
    passRef(templateRef: TemplateRef<any>, hcl: HeaderContainerLocation): void;
  }

```

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Програма та методика тестування

КПІ.ІП-4309.045440-05-51

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 року

ЗМІСТ

| | |
|------------------------------------|---|
| ОБ’ЄКТ ВИПРОБУВАНЬ | 6 |
| МЕТА ТЕСТУВАННЯ | 7 |
| МЕТОДИ ТЕСТУВАННЯ | 8 |
| ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ | 9 |

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-05-51 | Арк. |
| | | | | | | 5 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ОБ’ЄКТ ВИПРОБУВАНЬ

Об’єктом випробувань це веб-застосунок, для управління користувацькими коштами, розроблений за використанням фреймворку Angular.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-05-51 | Арк. |
| | | | | | | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- відповідність роботи описаного функціонала до функціональних вимог;
- відображення елементів інтерфейсу;
- стабільність роботи та працездатність функціоналу;
- відповідність додатку до описаних вимог;
- зручність та простота в використанні застосунку.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-05-51 | Арк. |
| | | | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

МЕТОДИ ТЕСТУВАННЯ

Для тестування застосунку обрано декілька підходів тестування:

- тестування користувацького інтерфейсу;
- тестування за допомогою порівняння вихідних даних - очікуваних та фактичних результатів.

Використовуються наступні методи тестування:

- тестування інтерфейсу за допомогою користування;
- функціональне тестування шляхом юніт тестування.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-05-51 | Арк. |
| | | | | | | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування функціонуванні частини виконується написанням unit тестів за допомогою вбудованого інструменту для написання тестів Karma та Jasmin, Unit тести описуються після виконання певного функціональної вимоги.

Працездатність застосунку тестується ручним тестуванням. Перевіряється граничні значення полів. Перевіряються основні функціональні вимоги.

Тестується коректність відображення даних на інтерфейсі.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-4309.045440-05-51 | Арк. |
| | | | | | | 9 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Керівництво користувача

КП.ІП-4309.045440-06-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 рік

КЕРІВНИЦТВО КОРИСТУВАЧА

Після переходу на головну сторінку сайту рисунок 1.1, користувачу надається можливість виконати авторизацію та реєстрацію.

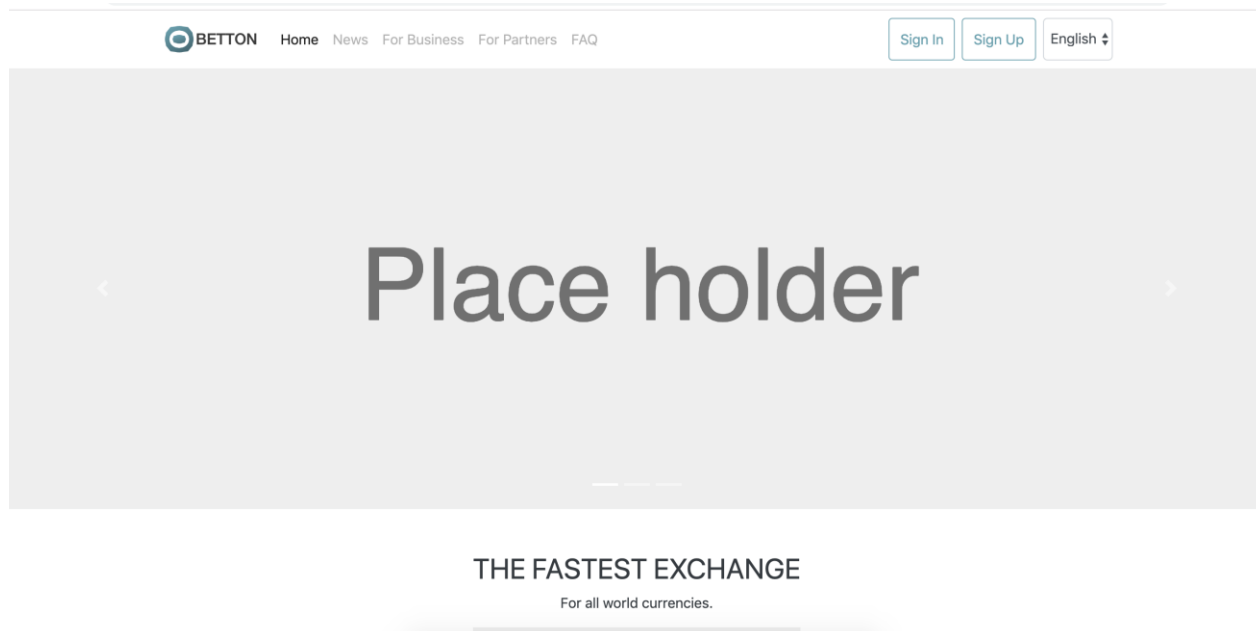


Рисунок 1.1 - Головна сторінка сайту

Після натиску кнопки вхід користувач переходить на сторінку авторизації чи за прямим посиланням РУТ/sign-in користувач за замовченням отримує форму вводу емейду та паролю рис. 1.2. Також альтернативою для входу надається форми входу через телефон ОТР(рис. 1.3) та QR(рис. 1.4) код. У разі збережених браузером креншалів для входу вони автоматично будуть заповнені як показано на рис. 1.2.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| | | | | | | 2 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

The screenshot shows the BETTON website header with navigation links (Home, News, For Business, For Partners, FAQ) and buttons for Sign In, Sign Up, and English. The main content area features a 'Sign In' form with three tabs: Email, Phone, and QR Code. The Email tab is active, showing a text input for 'Email (username@domain.com)' with the value 'yevhenii.kamenskyi@gmail.com', a password input field with masked characters and a toggle eye icon, and a 'forgot password?' link. A 'Sign in' button is at the bottom of the form.

Рисунок 1.2 - Сторінка авторизації имейл

Для входу через імейл потрібно заповнити поля імейл та поле пароль після натиснути кнопку “Вхід”. У разі некоректних даних система повідомить

The screenshot shows the BETTON website header with navigation links (Home, News, For Business, For Partners, FAQ) and buttons for Sign In, Sign Up, and English. The main content area features a 'Sign In' form with three tabs: Email, Phone, and QR Code. The Phone tab is active, showing a text input for 'Phone number (+380 000 00 00)' with a 'Send SMS' button. Below it is a 'Pin code' input field and a 'Sign in' button. A message 'SMS isn't work, call me instead!' is displayed.

про помилку.

Рисунок 1.3 - Сторінка авторизації телефон

Для входу через телефон потрібно заповнити поле “номер телефона” та ввести пінкот який прийде у вигляді СМС до вашого мобільного телефону. У разі відсутності СМС надають можливість повторити запит натиснувши на кнопку “Відправити СМС”.

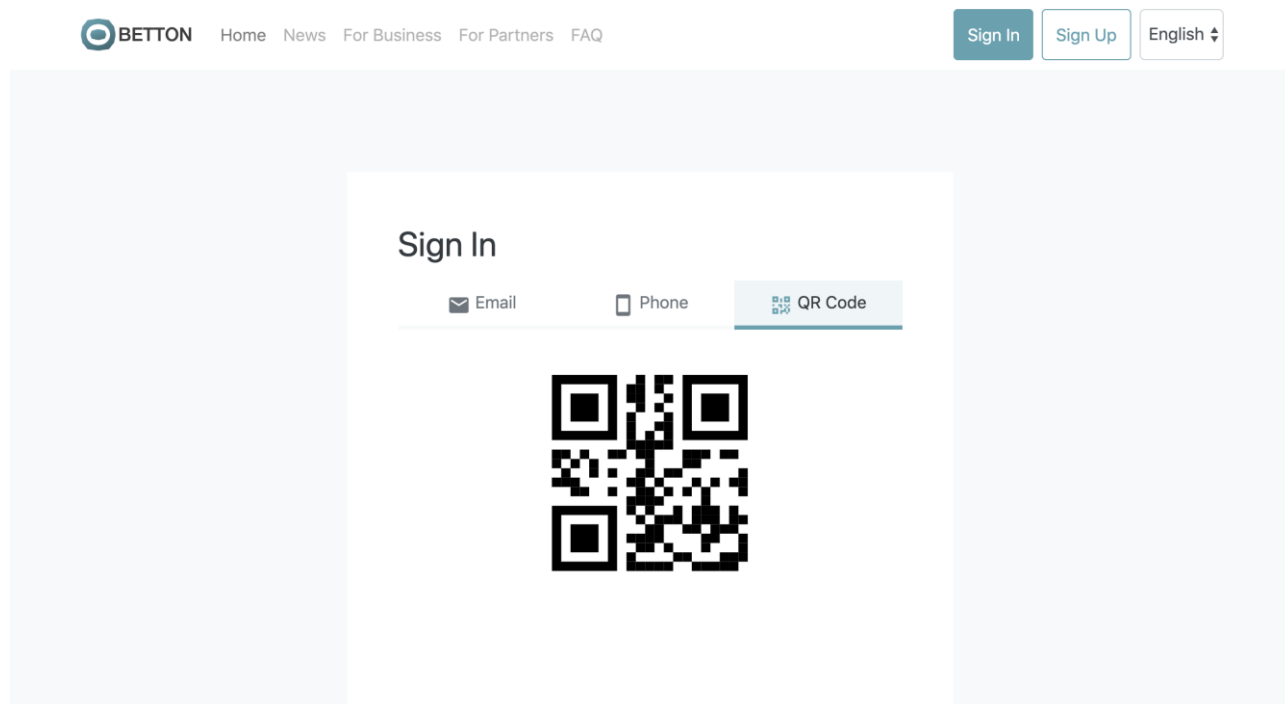


Рисунок 1.4 - Сторінка авторизації QR

У разі відсутності паролю чи втрати, користувачу надається механізм відновлення паролю.

Щоб відновити пароль потрібно перейти за посиланням РУТ/restore чи за натиснути “Забув пароль” на сторінці входу через імейл рис 1.5.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

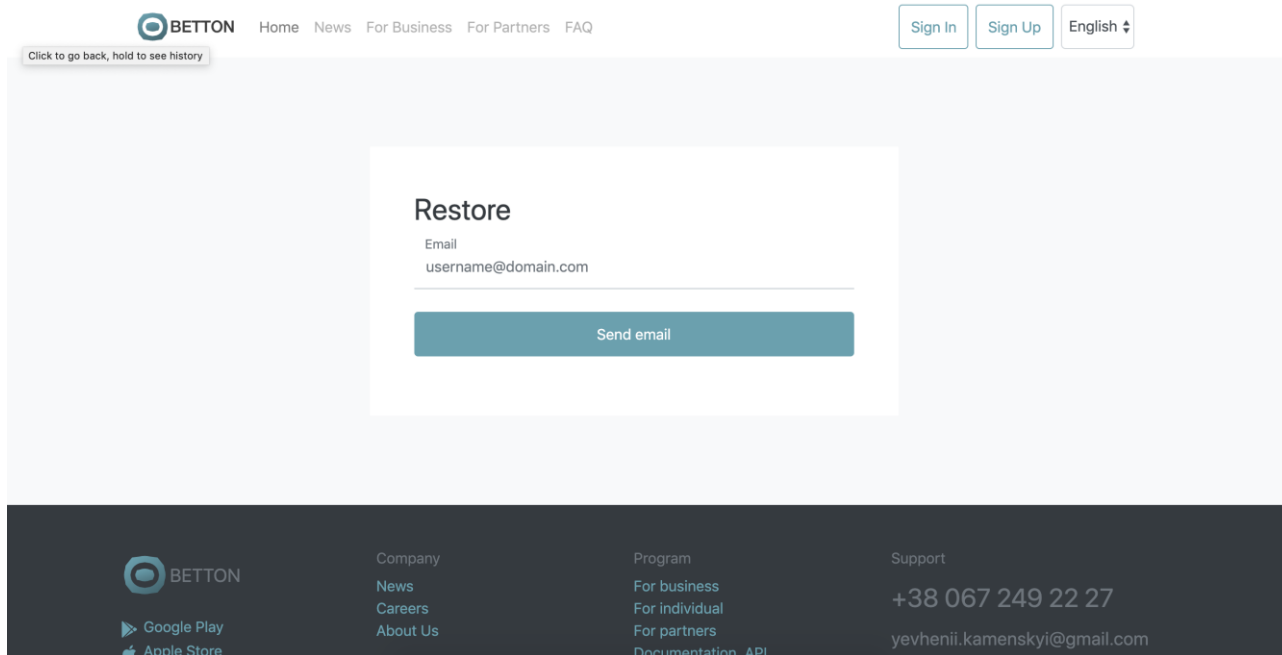
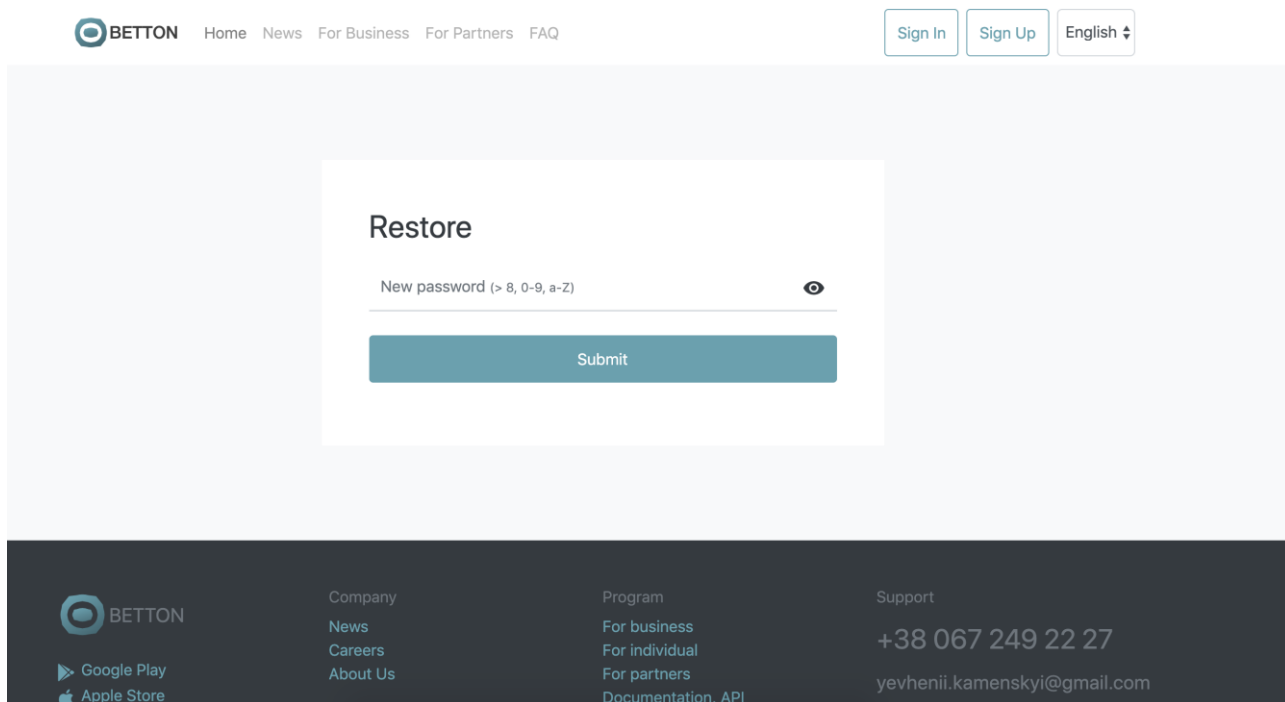


Рисунок 1.5 - Сторінка відновлення паролю

Ввести імейл користувача та перейти на пошту та там за посиланням



перейти на сторінку відновлення паролю рис 1.6.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 5 |

Рисунок 1.6 - Сторінка відновлення паролю

Після підтвердження нового паролю, автоматично система переведе на сторінку авторизації.

У разі створення нового користувача існує сторінка реєстрації рис 1.7.

Рисунок 1.7 - Сторінка реєстрації

Для реєстрації в системі необхідно заповнити поля імейл та пароль та надати згоду політиці конфіденційності.

Після авторизації та реєстрації користувач потрапляє на сторінку дашборду рис 1.8.

Рисунок 1.8 - Сторінка дашборду

Для отримання основної інформації існує сторінка “Дашборд” де розміщені наступні віджети:

- стан по рахункам - інформація про поточний стан активів користувача;
- останні транзакції - перелік останніх переведень по всіх рахункам;
- переведення - форма для швидкого переведення;
- обмін валюти - форма для швидкої покупки чи продажу валюти;
- платежі - зручні шаблони для оплати чи переведення коштів.

Віджет “Переведення” автоматично заповнюється, після підтвердження платежу відображається поп-ап рис 1.9.

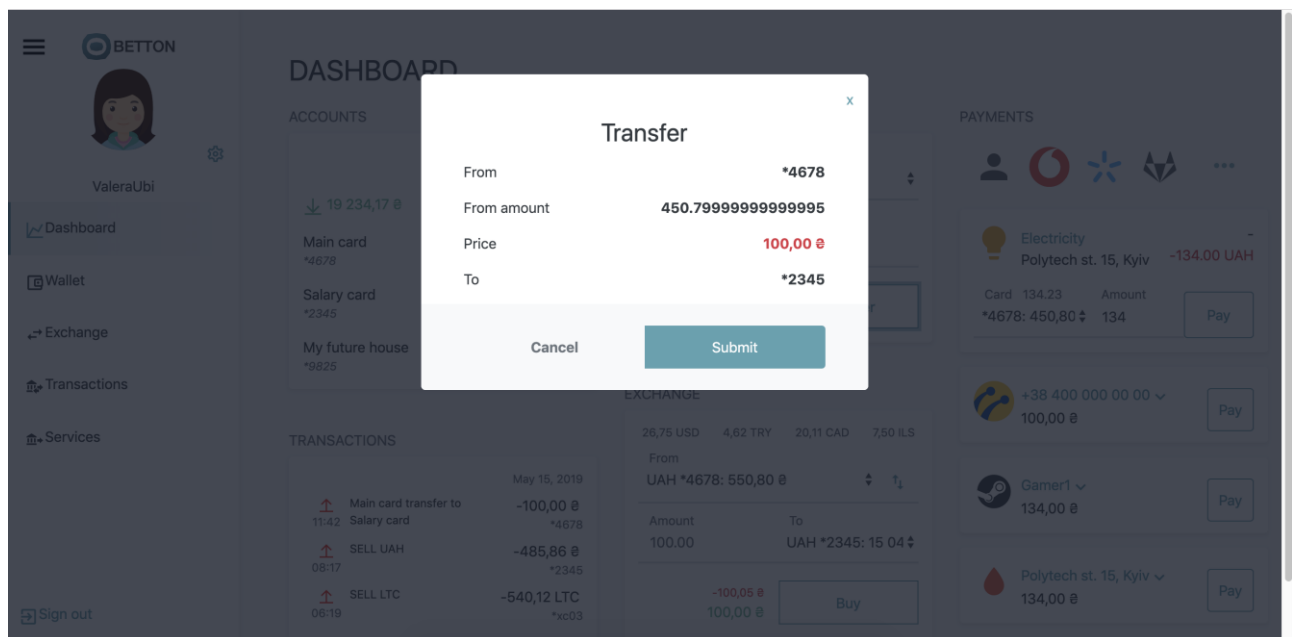


Рисунок 1.9 - Сторінка дашборду поп-ап

Сторінка гаманець відображає стан балансів по кожному рахунку та підводить тотожну по кожній валюті рис 1.10.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 7 |

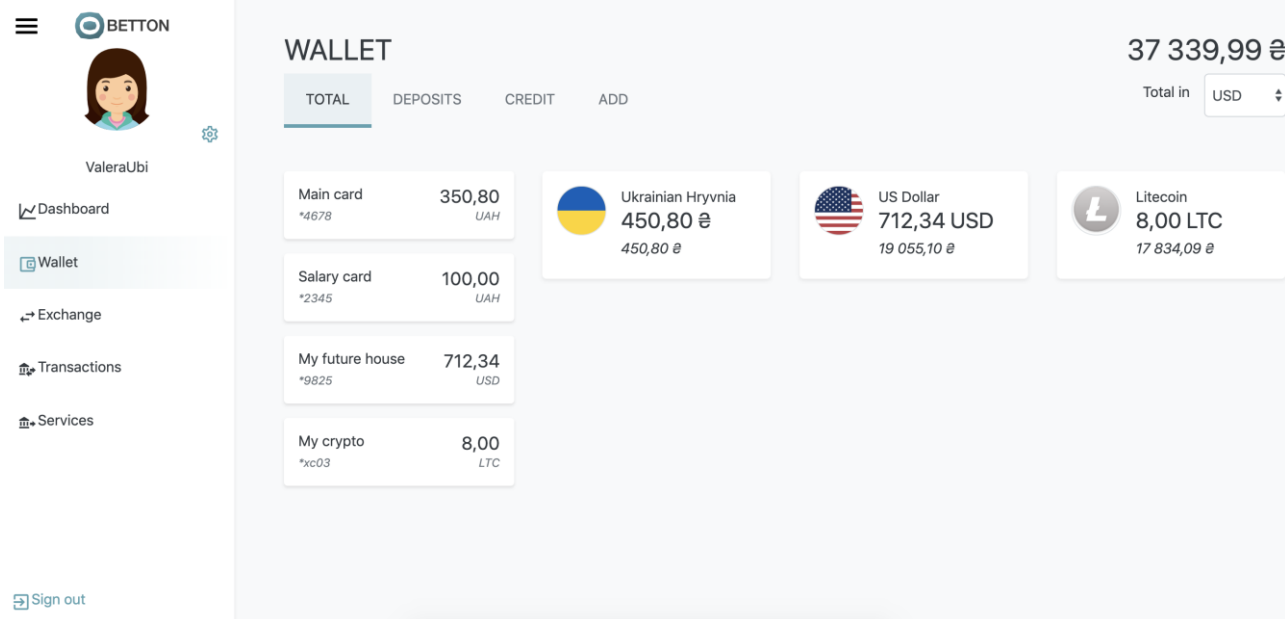


Рисунок 1.10 - Сторінка гаманець

Користувач може змінити його пріоритетну валюту у верхньому правому вуглі. Для отримання детальної інформації по кожному рахунку необхідно натиснути на карткове представлення рахунку.

На сторінці рахунку відображається детальна інформація по рахунку та основні операції по управліннями цими рахунками.

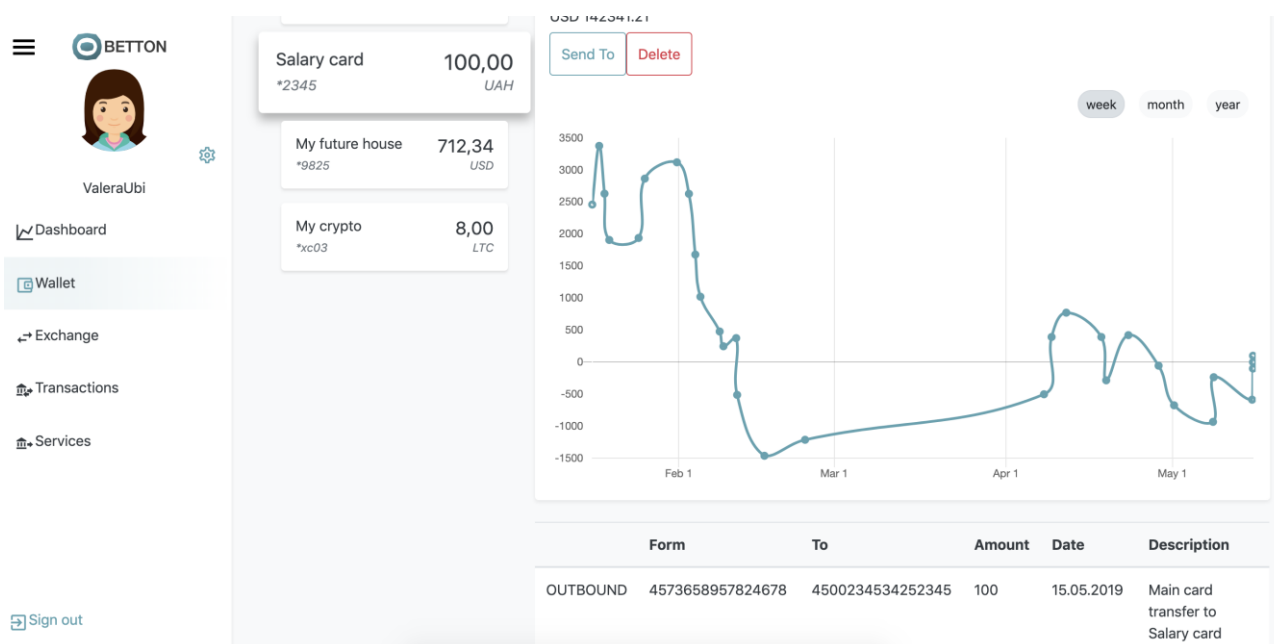


Рисунок 1.11 - Сторінка рахунку

Для додавання нового рахунку необхідно перейти на сторінку замовлення катки чи додавання існуючи рисунок 1.12 та рисунок 1.13.

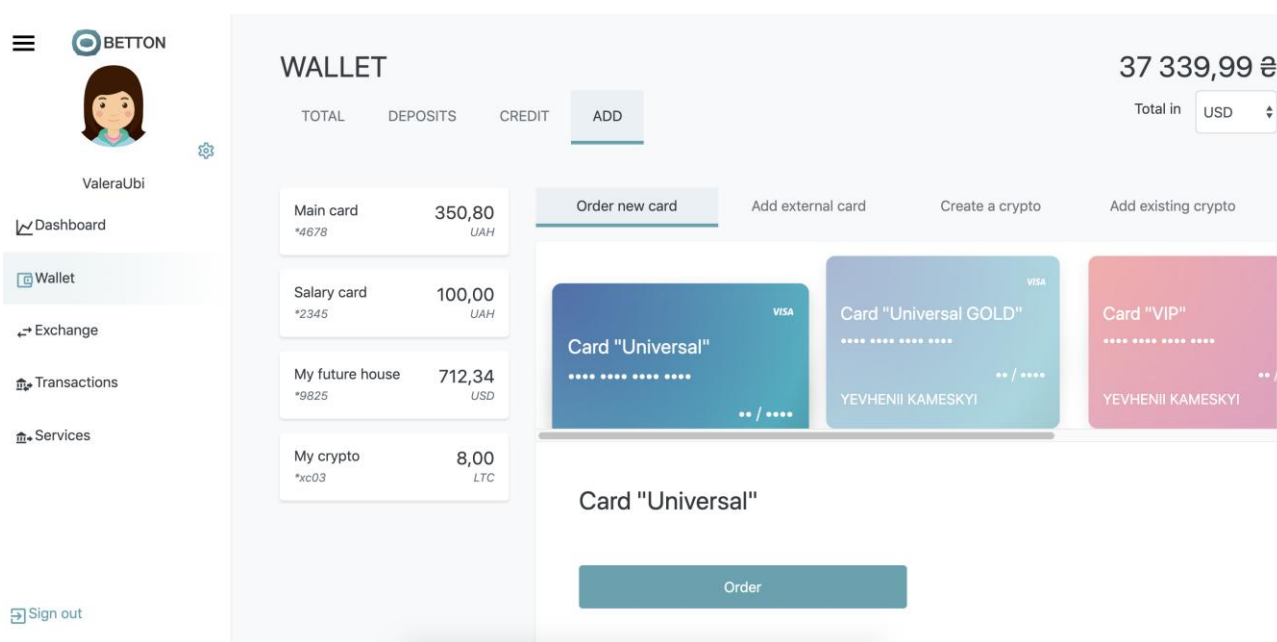


Рисунок 1.12 - Сторінка замовлення рахунку

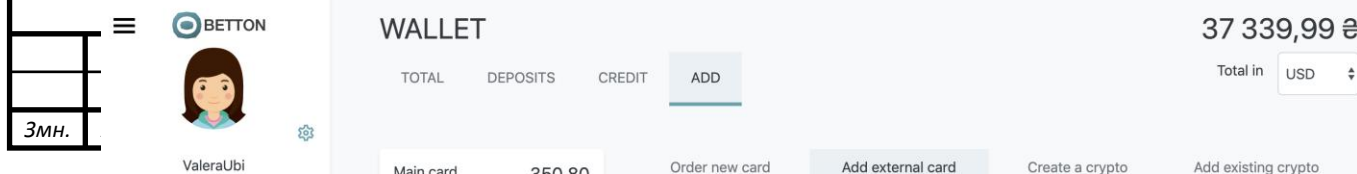


Рисунок 1.13 - Сторінка додавання картки

Після підтвердження рахунок автоматично створюється, та відображається форма успішного виконання рис 1.14.

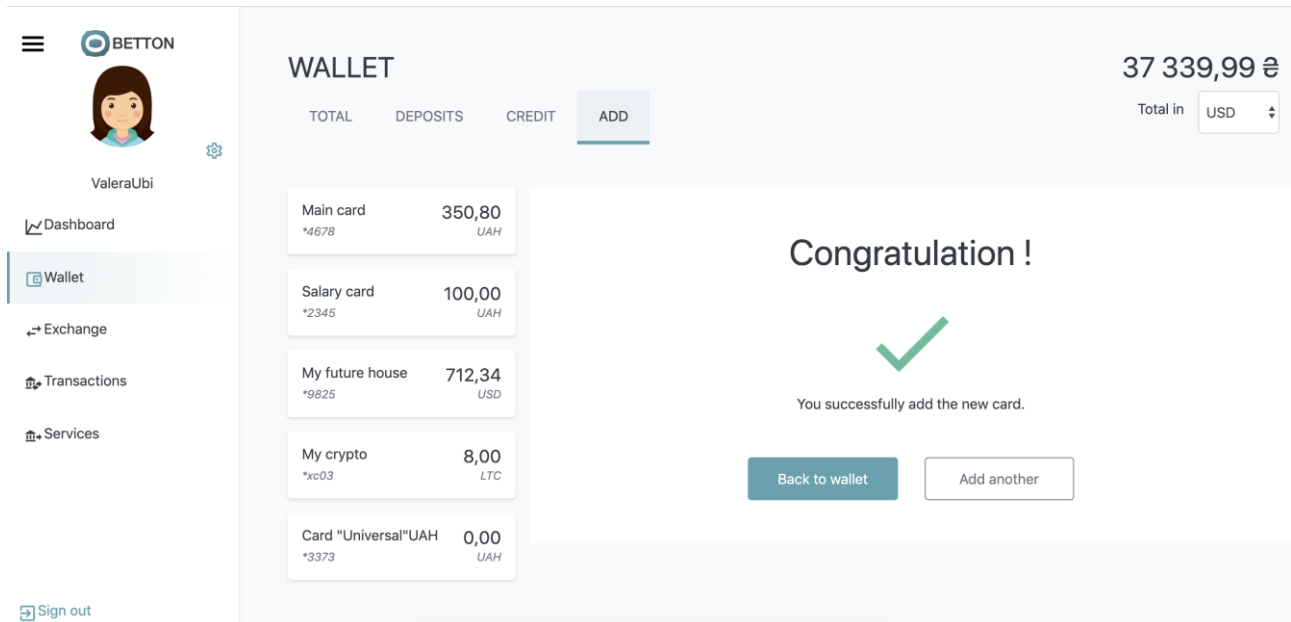
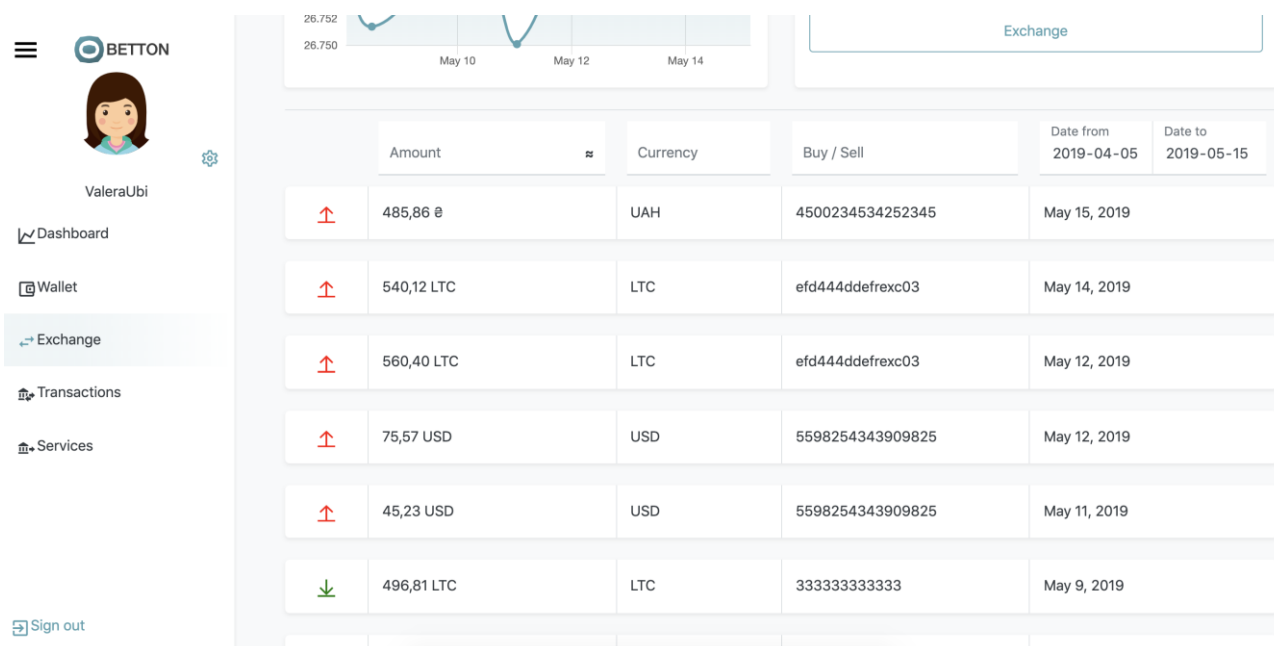


Рисунок 1.14 - Сторінка підтвердження

Сторінка обміну валют відображає курс відносно вибраних валют



користувачем та функціонал для купівлі чи протажу валюти рис. 1.15.

Рисунок 1.15 - Сторінка обміну валюти

На цій сторінці також розміщенна інформація про історію купівлі та продажу між різними рахунками рис 1.16.

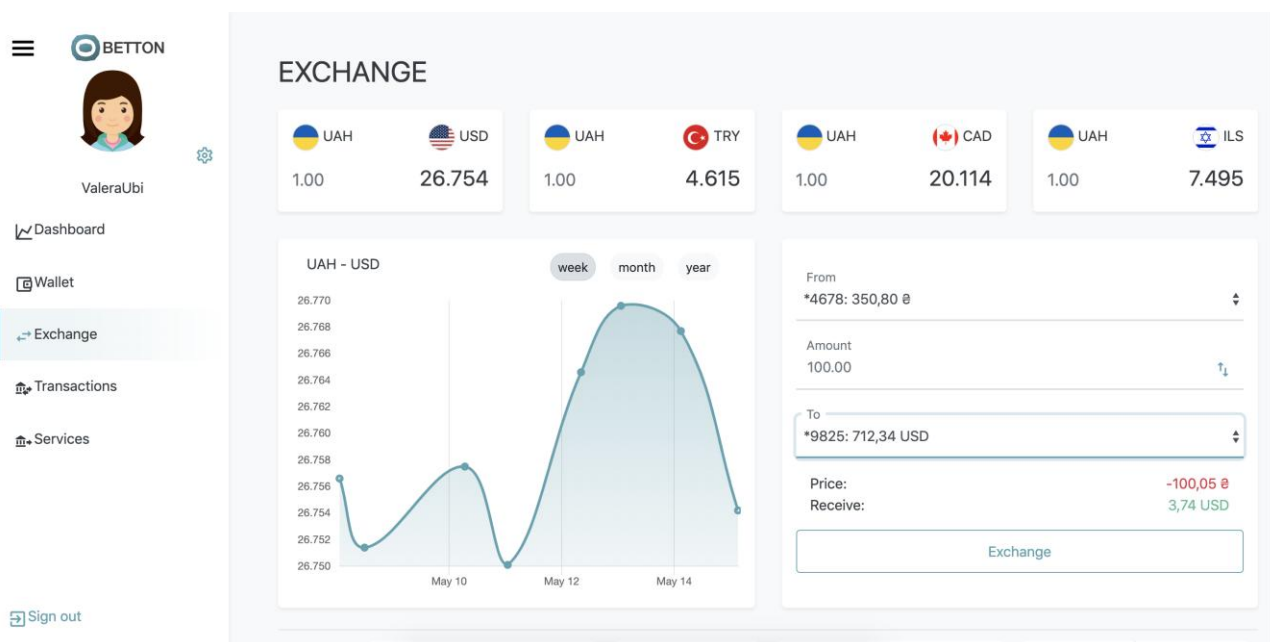


Рисунок 1.16 - Сторінка обміну валюти

Для переведення між рахунками використовують сторінку переводень
рис. 1.17.

BETTON

ValeraUbi

- Dashboard
- Wallet
- Exchange
- Transactions**
- Services
- Sign out

TRANSACTIONS

Account

Account name

From

- *4678: 550,80 ₪
- *2345: 15 046,87 ₪
- *9825: 712,34 USD
- *xc03: 8,00 LTC

Amount

100

Transfer

Card

Receiver account

Receiver

4500234534252345

price

total

Amount ≈ Form To Date from yyyy-mm-dd Date to yyyy-mm-dd

Рисунок 1.17 - Сторінка переводень



ValeraUbi

Dashboard

Wallet

Exchange

Transactions

Services

Sign out

SERVICES

Search Q

Place holder

Place holder

Place holder

Place holder

Place holder

Place holder

Mobile operator

Internet

Games

Mobile number
+380672492227Amount
0.01

Account

Pay

Для оплати платежів використовують сторінку сервісів рис. 1.18.

Рисунок 1.18 - Сторінка сервісів

У разі необхідності змінити користувацькі данні використовують сторінку налаштування рис 1.19.

Рисунок 1.19 - Сторінка налаштування



ValeraUbi

Dashboard

Wallet

Exchange

Transactions

Services

Sign out

SETTINGS

ACCOUNT

SECURITY

NOTIFICATION

Password

Change password

Email
valeraFed@gmail.com

Change email

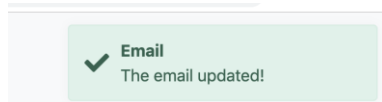
Phone number
+380673542323

Change phone number

Рисунок 1.20 - Сторінка безпека

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 13 |

На даній сторінці (Сторінка безпеки) користувач має можливість змінити пароль, імейл та номер телефону рис. 1.20. Після зміни імейлу користувачу



відображається повідомлення подібно рис 1.21.

Рисунок 1.21 - Повідомлення про зміну імейлу

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-06-34 | Арк. |
| | | | | | | 14 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Керівництво системного програміста

КП.ІП-4309.045440-08-32

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 рік

КЕРІВНИЦТВО СИСТЕМНОГО ПРОГРАМІСТА

Приклад конфігурації для серверу:

```
server {  
    listen 80;  
    listen 443 ssl;  
    ssl_certificate /etc/nginx/sites-enabled/certificate.pem;  
    ssl_certificate_key /etc/nginx/sites-enabled/private.key;  
    server_name fe;  
    location / {  
        root /var/www/fe;  
        try_files $uri $uri/ /index.html =404;  
    }  
    location /api/v1 {  
        proxy_pass https://localhost:9043/api/v1;  
        proxy_ssl_verify off;  
        proxy_redirect off;  
    }  
}
```

Сертифіки ssl :

```
ssl_certificate /etc/nginx/sites-enabled/certificate.pem;  
ssl_certificate_key /etc/nginx/sites-enabled/private.key;
```

Розміщення серверного застосунку:

```
proxy_pass https://localhost:9043/api/v1;
```

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-08-32 | Арк. |
| | | | | | | 2 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Розгортання застосунку зображено на рисунку 1.1.

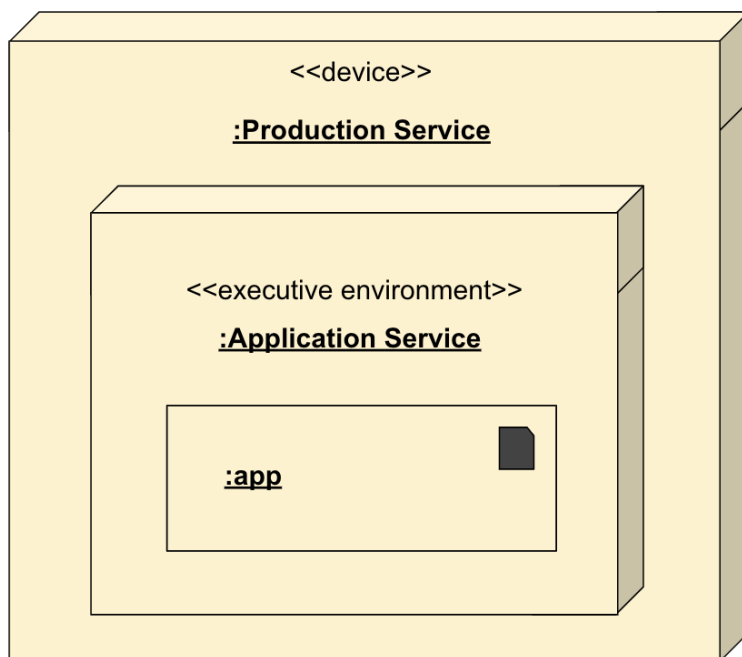


Рисунок 1.1 - Діаграма розгортання

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Керівництво програміста

КП.ІП-4309.045440-07-33

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 рік

КЕРІВНИЦТВО ПРОГРАМІСТА

Початок роботи

Для початку розробки проекту його необхідно скачати до локального репозиторію виконати команду - “git clone git@gitlab.com:kaskyi/bank-fe.git”.

```
Kamenskys-MacBook-Pro:tmp kamensky$ git clone git@gitlab.com:kaskyi/bank-fe.git
Cloning into 'bank-fe'...
remote: Enumerating objects: 1479, done.
remote: Counting objects: 100% (1479/1479), done.
remote: Compressing objects: 100% (774/774), done.
remote: Total 1479 (delta 807), reused 1286 (delta 660)
Receiving objects: 100% (1479/1479), 541.96 KiB | 880.00 KiB/s, done.
Resolving deltas: 100% (807/807), done.
Kamenskys-MacBook-Pro:tmp kamensky$
```

Рисунок 1.1 - Приклад виконання програми

Після необхідно перейти до дерикторії проекту - “cd bank-fe” та підтянути необхідні пакети для цього виконати команду : “npm install” чи “yarn install”
 рисунок 1.2

Рисунок 1.2 - Приклад виконання програми

```
Kamenskys-MacBook-Pro:bank-fe kamensky$ yarn install
yarn install v1.13.0
warning package-lock.json found. Your project contains lock files generated by tools other than Yarn. It is advised not to mix package manage
rs in order to avoid resolution inconsistencies caused by unsynchronized lock files. To clear this warning, remove package-lock.json.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning " > bootstrap@4.3.1" has unmet peer dependency "jquery@1.9.1 - 3".
warning " > bootstrap@4.3.1" has unmet peer dependency "popper.js@^1.14.7".
warning " > tsickle@0.35.0" has incorrect peer dependency "typescript@~3.4.1".
[4/4] Building fresh packages...
success Saved lockfile.
warning Your current version of Yarn is out of date. The latest version is "1.16.0", while you're on "1.13.0".
info To upgrade, run the following command:
$ brew upgrade yarn
Done in 23.55s.
```

Сервер розробки

Використовуйте одну з нижченаведених команд для запуску сервера розробки рисунок 1.3.

- запусити `npm run start` - сервер з проксі;
- запусити `npm run start: ssl` - сервер https з проксі;
- запусити `npm run start: noreload` - сервер з проксі і без автоматичного перезавантаження;

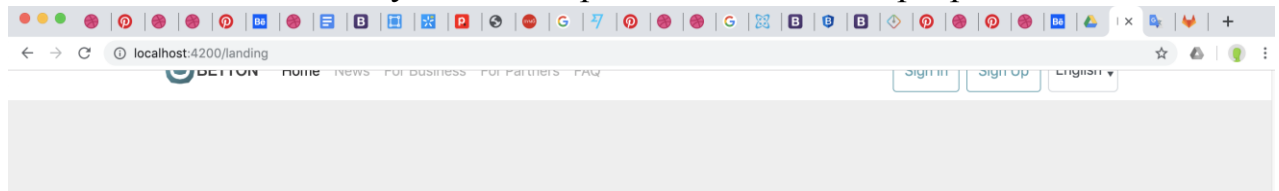
- запустіть ``npm run start: host`` - сервер стає доступним у вашій локальній мережі (ви можете використовувати його для тестування на гаджетах);

```
chunk {main} main.js, main.js.map (main) 656 kB [initial] [rendered]
chunk {pages-account-account-module} pages-account-account-module.js, pages-account-account-module.js.map (pages-account-account-module) 42.9 kB [rendered]
chunk {pages-accounts-accounts-module} pages-accounts-accounts-module.js, pages-accounts-accounts-module.js.map (pages-accounts-accounts-module) 107 kB [rendered]
chunk {pages-dashboard-dashboard-module} pages-dashboard-dashboard-module.js, pages-dashboard-dashboard-module.js.map (pages-dashboard-dashboard-module) 103 kB [rendered]
chunk {pages-exchange-exchange-module} pages-exchange-exchange-module.js, pages-exchange-exchange-module.js.map (pages-exchange-exchange-module) 37.7 kB [rendered]
chunk {pages-landing-landing-module} pages-landing-landing-module.js, pages-landing-landing-module.js.map (pages-landing-landing-module) 11.1 kB [rendered]
chunk {pages-plugins-plugins-module} pages-plugins-plugins-module.js, pages-plugins-plugins-module.js.map (pages-plugins-plugins-module) 7.25 kB [rendered]
chunk {pages-restore-restore-module} pages-restore-restore-module.js, pages-restore-restore-module.js.map (pages-restore-restore-module) 24.8 kB [rendered]
chunk {pages-services-services-module} pages-services-services-module.js, pages-services-services-module.js.map (pages-services-services-module) 17.2 kB [rendered]
chunk {pages-sign-in-sign-in-module} pages-sign-in-sign-in-module.js, pages-sign-in-sign-in-module.js.map (pages-sign-in-sign-in-module) 37.5 kB [rendered]
chunk {pages-sign-up-sign-up-module} pages-sign-up-sign-up-module.js, pages-sign-up-sign-up-module.js.map (pages-sign-up-sign-up-module) 18.8 kB [rendered]
chunk {pages-transactions-transactions-module} pages-transactions-transactions-module.js, pages-transactions-transactions-module.js.map (pages-transactions-transactions-module) 36.7 kB [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 248 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 9.6 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 954 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.55 MB [initial] [rendered]
[wdm]: Compiled successfully.
```

Рисунок 1.3 - Приклад виконання програми

Після успішного запуску програми перейдіть до ``http: // localhost: 4200 /`` у браузері рисунок 1.4.

Рисунок 1.4 - Приклад виконання програми



Кодекс

Запустіть ``ng generate компонент-ім'я компонента``, щоб створити новий компонент. Ви також можете використовувати ``ng generate Directive | pipe | service | class | module |``.

Збірка

Запустіть ``npm run build``, щоб створити проект. Артефакти збирання будуть зберігатися в каталозі ``dist /``.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-07-33 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

Використовуйте ``npm run build: prod`` для виробничої збірки.

Аналіз зв'язків

Запустіть `"npm run build"`, щоб створити звіт про пакет.

Виконайте ``report: bundle`` для візуалізації розміру вихідних файлів webpack.

Виконання lint

Перевірка написання коду на загальні конвенції.

Запустіть ``npm run lint``, щоб виконати аналіз статичного коду.

Запустіть ``npm run lint: ci``, щоб створити звіти про складки.

Виконання тестів

Запустіть тест ``npm run ``, щоб виконати модульні тести через [Karma] (<https://karma-runner.github.io>).

Запустіть ``npm run test:ci``, щоб виконати модульні тести один раз і створити звіт про покриття коду.

Виконання розподільних тестів

Запустіть ``ng e2e``, щоб виконати наскрізні тести через [Protractor] (<http://www.protractortest.org/>).

Перед запуском тестів переконайтеся, що ви обслуговуєте додаток через ``npm run start``.

Формат фіксування змін

Перед початком внесення змін переконайтеся, що він успішно виконав команди ``lint`` та ``test``.

Будь-яке повідомлення про фіксацію повинно підтримувати [Формат] (<https://github.com/angular/angular/blob/master/CONTRIBUTING.md#commit-message-format>).

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | КПІ.ІП-4309. 045440-07-33 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування для управління коштами

Графічний матеріал

КП.ІП-4309.045440-09-99

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.К. Очеретяний

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Є.А. Каменський

Київ – 2019 рік

